

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2007-052790

(43)Date of publication of application : 01.03.2007

(51)Int.Cl.

G06F 15/80 (2006.01)

(21)Application number : 2006-221861

(71)Applicant : INTERNATL BUSINESS MACH  
CORP <IBM>

(22)Date of filing : 16.08.2006

(72)Inventor : JOHNS CHARLES RAY  
SWANSON TODD  
LIU PEICHUN PETER  
DAY MICHAEL NORMAN  
TRUONG THUONG QUANG

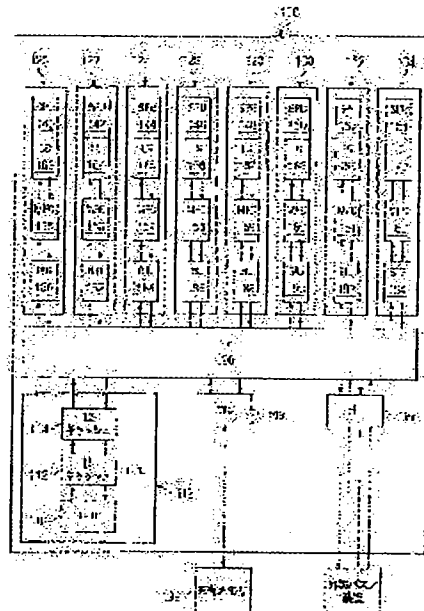
(30)Priority

Priority number : 2005 207986 Priority date : 19.08.2005 Priority country : US

(54) SYSTEM, METHOD, COMPUTER PROGRAM AND DEVICE FOR COMMUNICATING  
COMMAND PARAMETER BETWEEN PROCESSOR AND MEMORY FLOW CONTROLLER

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a system and a method for communicating command parameters between a processor and a memory flow controller.  
**SOLUTION:** This application utilizes a channel interface as a main mechanism for communication between the processor and the memory flow controller. The channel interface provides a channel for executing communication with, for instance, a processor facility, a memory flow control facility, a machine status register, and an external processor interrupt facility. When data to be read from a corresponding register by a blocking channel are not usable or there is no writing space in the corresponding register, the processor is brought into a low-power "stall" state. When the data are made usable or a space is released, the processor is automatically called via communication on the blocking channel.



(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2007-52790

(P2007-52790A)

(43) 公開日 平成19年3月1日(2007.3.1)

(51) Int. Cl.

G06F 15/80

(2006.01)

F I

G06F 15/80

テーマコード (参考)

審査請求 未請求 請求項の数 24 OL (全 71 頁)

(21) 出願番号 特願2006-221861 (P2006-221861)  
 (22) 出願日 平成18年8月16日 (2006.8.16)  
 (31) 優先権主張番号 11/207986  
 (32) 優先日 平成17年8月19日 (2005.8.19)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 390009531  
 インターナショナル・ビジネス・マシー  
 ズ・コーポレーション  
 INTERNATIONAL BUSIN  
 ESS MACHINES CORPO  
 RATION  
 アメリカ合衆国10504 ニューヨーク  
 州 アーモンク ニュー オーチャード  
 ロード  
 (74) 代理人 100108501  
 弁理士 上野 剛史  
 (74) 代理人 100112690  
 弁理士 太佐 穂一  
 (74) 代理人 100091568  
 弁理士 市位 嘉宏

最終頁に続く

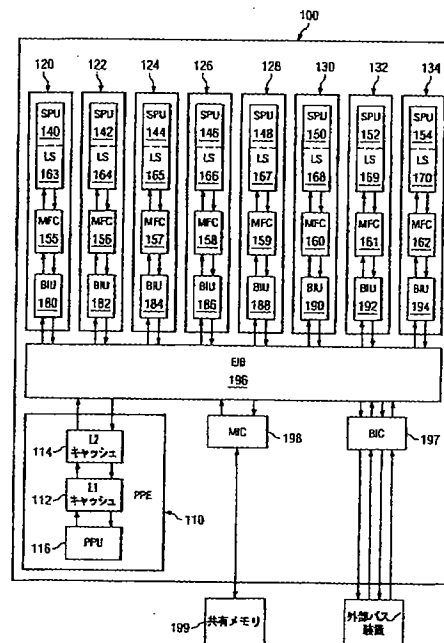
(54) 【発明の名称】 プロセッサとメモリ・フロー・コントローラとの間でコマンド・パラメータを通信するためのシステム、方法、コンピュータ・プログラム、および装置

(57) 【要約】 (修正有)

【課題】 プロセッサとメモリ・フロー・コントローラとの間でコマンド・パラメータを通信するためのシステムおよび方法を提供する。

【解決手段】 本願は、チャンネル・インターフェースを、プロセッサとメモリ・フロー・コントローラとの間の通信のための主要な機構として活用する。チャンネル・インターフェースは、例えば、プロセッサ・ファシリティ、メモリ・フロー制御ファシリティ、マシン・ステータス・レジスタ、および外部プロセッサ割り込みファシリティと通信するためのチャンネルを提供する。ブロッキング・チャンネルによって、対応レジスタから読み出されるべきデータが使用可能でない場合、または対応レジスタに書き込む空間がない場合には、プロセッサは、低電力の「ストール」状態にされる。データが使用可能になるか、または空間が開放されると、プロセッサは、ブロッキング・チャンネル上での通信を介して自動的に呼び起こされる。

【選択図】 図1



## 【特許請求の範囲】

## 【請求項 1】

データ処理システムにおいて、プロセッサとメモリ・フロー制御（MFC）ユニットとの間で命令およびデータを通信するための方法であって、

前記プロセッサと前記MFCユニットとの間でデータを転送するために、前記プロセッサと前記MFCユニットとの間で1つ以上のMFCコマンド・パラメータ・チャンネルを確立するステップと、

前記MFCユニットにおいて、前記1つ以上のMFCコマンド・パラメータ・チャンネルを介してMFCコマンド・パラメータ・データを受信するステップと、

前記1つ以上のチャンネルに関連した1つ以上のMFCコマンド・パラメータ・レジスタに前記データを記憶するステップと、

前記1つ以上のMFCコマンド・パラメータ・レジスタに記憶された前記データに基づいて、MFCコマンド・キュー内の1つ以上のコマンドを処理するステップとを備える、方法。

## 【請求項 2】

前記MFCコマンド・パラメータ・データは、マルチプロセッサ・システム内のプロセッサから受信される、請求項 1 に記載の方法。

## 【請求項 3】

前記プロセッサは、前記プロセッサとローカル記憶装置とを有する相乗作用処理ユニットの一部である、請求項 2 に記載の方法。

## 【請求項 4】

前記1つ以上のMFCコマンド・パラメータ・チャンネルは、非ブロッキング・チャンネルであり、前記1つ以上のMFCコマンド・パラメータ・チャンネルは、チャンネル・カウントを有しない、請求項 2 に記載の方法。

## 【請求項 5】

前記1つ以上のMFCコマンド・パラメータ・チャンネルは、MFCローカル記憶アドレス・チャンネルと、MFC有効アドレスhighチャンネルと、MFC有効アドレスlowまたはリスト・アドレス・チャンネルと、MFC転送サイズまたはリスト・サイズ・チャンネルと、MFCコマンド・タグ識別チャンネルと、MFCコマンド操作符号チャンネルと、MFCクラスIDチャンネルとのうちの少なくとも1つを備える、請求項 1 に記載の方法。

## 【請求項 6】

前記1つ以上のMFCコマンド・パラメータ・チャンネルは、MFCコマンド操作符号チャンネルを備え、前記MFCコマンド・パラメータ・データは、関連MFCコマンドによって実行されるべき動作を識別する、前記MFCコマンド操作符号チャンネル上で送信されたコマンド操作符号を含む、請求項 1 に記載の方法。

## 【請求項 7】

前記コマンド操作符号の有効性を非同期にチェックするステップと、

前記コマンド操作符号が無効である場合に、MFCコマンド・キューの処理を中断するステップと、

無効のMFCコマンド割り込みを生成するステップとをさらに備える、請求項 6 に記載の方法。

## 【請求項 8】

前記1つ以上のMFCコマンド・パラメータ・チャンネルは、MFCコマンド・クラスIDチャンネルを備え、前記MFCコマンド・パラメータ・データは、関連MFCコマンドのキャッシュ置換を制御するために置換管理テーブルと共に使用される、前記MFCコマンド・クラスIDチャンネル上で受信された置換クラスIDを含む、請求項 1 に記載の方法。

## 【請求項 9】

前記置換クラスIDは、制御プロセッサ動作のためのロード・記憶アドレスから生成さ

10

20

30

40

50

れる、請求項 8 に記載の方法。

【請求項 10】

前記置換クラス ID に基づいて、前記置換管理テーブルのインデックスを生成するステップと、

前記インデックスによって識別された前記置換管理テーブルの入力に基づいて、置換ポリシーを制御するステップと

をさらに備える、請求項 8 に記載の方法。

【請求項 11】

前記 MFC コマンド・パラメータ・データは、記憶に対するアクセスを識別するために使用される、前記 MFC コマンド・クラス ID チャンネル上で受信された転送クラス ID をさらに含む、請求項 8 に記載の方法。 10

【請求項 12】

前記 1 つ以上の MFC コマンド・パラメータ・チャンネルは、MFC コマンド・タグ識別チャンネルを備え、前記 MFC コマンド・パラメータ・データは、MFC コマンドの群を識別する、前記 MFC コマンド・タグ識別チャンネル上で受信されたコマンド・タグ識別子を含む、請求項 1 に記載の方法。

【請求項 13】

前記コマンド・タグ識別子の有効性を非同期にチェックするステップと、

前記コマンド・タグ識別子が 0 に設定されていない場合に、前記 MFC コマンド・キューの処理を中断するステップと、 20

割り込みを生成するステップと

をさらに備える、請求項 12 に記載の方法。

【請求項 14】

前記 1 つ以上の MFC コマンド・パラメータ・チャンネルは、MFC コマンド転送サイズまたはリスト・サイズ・チャンネルを備え、前記 MFC コマンド・パラメータ・データは、関連 MFC コマンドの直接メモリ・アドレス (DMA) 転送のサイズを識別する、前記 MFC コマンド転送サイズまたはリスト・サイズ・チャンネル上で受信された転送サイズまたはリスト・サイズ・パラメータを含む、請求項 1 に記載の方法。

【請求項 15】

前記転送サイズまたはリスト・サイズ・パラメータの有効性を非同期に検証するステップと、 30

前記転送サイズまたはリスト・サイズ・パラメータが無効である場合に、MFC コマンド・キューの処理を中断するステップと、

MFC・DMA アラインメント割り込みを生成するステップと

をさらに備える、請求項 14 に記載の方法。

【請求項 16】

前記 1 つ以上の MFC コマンド・パラメータ・チャンネルは、MFC ローカル記憶アドレス・チャンネルを備え、前記 MFC コマンド・パラメータ・データは、MFC 転送の送信元または送信先アドレスのいずれかを識別する、前記 MFC ローカル記憶アドレス・チャンネル上で受信された前記 MFC コマンドに関連した SPU ローカル記憶アドレス・パラメータを含む、請求項 1 に記載の方法。 40

【請求項 17】

前記 SPU ローカル記憶アドレス・パラメータの有効性を非同期で検証するステップと、

前記 SPU ローカル記憶アドレスが整列していない場合に、MFC コマンド・キューの処理を中断するステップと、

MFC・DMA アラインメント例外を生成するステップと

をさらに備える、請求項 16 に記載の方法。

【請求項 18】

前記 1 つ以上の MFC コマンド・パラメータ・チャンネルは、MFC 有効アドレス 10 50

wまたはリスト・アドレス・チャンネルを備え、前記MFCコマンド・パラメータ・データは、MFC有効アドレスlowまたはリスト・アドレス・チャンネル上で受信された関連MFCコマンドについてのMFC有効アドレスlowまたはリスト・アドレスを含む、請求項1に記載の方法。

【請求項19】

MFC有効アドレスlowまたはリスト・アドレスの有効性を非同期に検証するステップと、

前記MFC有効アドレスlowまたはリスト・アドレスが整列していない場合に、MFCコマンド・キューの処理を中断するステップと、

割り込みを生成するステップと

10

をさらに備える、請求項18に記載の方法。

【請求項20】

前記1つ以上のMFCコマンド・パラメータ・チャンネルは、MFC有効アドレスhighチャンネルを備え、前記MFCコマンド・パラメータ・データは、MFC有効アドレスhighチャンネル上で受信された関連MFCコマンドについてのMFC有効アドレスhighアドレスを含む、請求項1に記載の方法。

【請求項21】

MFC有効highアドレスの有効性を非同期に検証するステップと、

前記MFC有効highアドレスが整列していない場合に、MFCコマンド・キューの処理を中断するステップと、

20

割り込みを生成するステップと

をさらに備える、請求項20に記載の方法。

【請求項22】

前記MFCコマンド操作符号チャンネルへのチャンネル書き込み命令を受信するステップと、

前記MFCコマンド操作符号チャンネルへの前記チャンネル書き込み命令に応答して、前記1つ以上のMFCコマンド・パラメータ・チャンネルに保持されたMFCコマンド・パラメータをMFCコマンド・キューへ送るステップと

をさらに備える、請求項6に記載の方法。

【請求項23】

30

コンピュータ読み取り可能なプログラムを有するコンピュータ使用可能な媒体を備えるシステムであって、前記コンピュータ読み取り可能なプログラムは、コンピューティング装置上で実行されると前記コンピューティング装置に対して、

プロセッサとMFCユニットとの間でデータを転送するのに必要な、前記プロセッサと前記MFCユニットとの間の1つ以上のMFCコマンド・パラメータ・チャンネルを確立するステップと、

前記MFCユニットにおいて、前記1つ以上のMFCコマンド・パラメータ・チャンネルを介してMFCコマンド・パラメータ・データを受信するステップと、

前記1つ以上のチャンネルに関連した1つ以上のMFCコマンド・パラメータ・レジスタにデータを記憶するステップと、

40

前記1つ以上のMFCコマンド・パラメータ・レジスタに記憶された前記データに基づいて、MFCコマンド・キュー内の1つ以上のコマンドを処理するステップとを行わせる、システム。

【請求項24】

プロセッサと、

プロセッサに結合されたメモリであって、前記プロセッサによって実行されると前記プロセッサに対して、

前記プロセッサとMFCユニットとの間でデータを転送するのに必要な、前記プロセッサと前記MFCユニットとの間の1つ以上のMFCコマンド・パラメータ・チャンネルを確立するステップと、

50

前記MFCユニットにおいて、前記1つ以上のMFCコマンド・パラメータ・チャンネルを介してMFCコマンド・パラメータ・データを受信するステップと、

前記1つ以上のチャンネルに関連した1つ以上のMFCコマンド・パラメータ・レジスタにデータを記憶するステップと、

前記1つ以上のMFCコマンド・パラメータ・レジスタに記憶された前記データに基づいて、MFCコマンド・キュー内の1つ以上のコマンドを処理するステップと  
を行わせる命令を含むメモリと  
を備える、装置。

【発明の詳細な説明】

【技術分野】

10

【0001】

本発明は、一般的には、改良されたデータ処理システムおよび方法に関する。より特定のには、本発明は、プロセッサとメモリ・フロー・コントローラとの間でコマンド・パラメータを通信するためのシステムおよび方法に向けられている。

【背景技術】

【0002】

通常、従来技術においては、CPUまたは他の処理ユニット(PU)がプログラム外部の何らかのイベントを待っている場合には、オペレーティング・システムまたはアクティブなプログラムは、プログラムに関連してPUによって使用されているイベント・レジスタを読み続けるようなポーリング・ループを、待っているイベントが生じるまで実行することになる。プログラムが動作している間は、PUはイベント・レジスタをポーリングし続け、有用なワークを行っていない。

20

【発明の開示】

【発明が解決しようとする課題】

【0003】

典型的な現在のプロセッサは、この通信のために、仮想メモリと、外部装置のメモリ・マッピングとを使用することが多い。その一方で、プロセッサによっては、特にマルチプロセッサ環境においては、ローカル・メモリにしかアクセスできず、仮想メモリにアクセスできないものもある。ローカル・メモリには限界があり、典型的なマルチプロセッサ構成においては、このローカル・メモリの外部のメモリは、ロード・記憶動作によってアクセスすることができない。よって、外部装置からの応答を待っている間は、他のPU機能についてのローカル・メモリの使用は制限されている。PUがいくつかの装置からの通信応答を同時に待っている場合には、他の機能についての使用可能なメモリは、さらに制限されている。

30

【0004】

また、メモリは、入力レジスタまたは出力レジスタ内に有効データがあるかどうかを常に把握するために使用されてもよい。有効データとは、受信装置によって使用されるレジスタ内にあるが受信装置によってまだアクセスされていないデータのことである。よって、上記から、最先端のコンピューティング装置におけるメモリ・リソースの流出の原因となるものが数多くあることは明らかである。

40

【0005】

上記に鑑み、プロセッサと、プロセッサの外部にある他の装置、例えば、他のプロセッサ、入出力(I/O)装置などとの間の通信のためのシステムおよび方法であって、そのような通信がプロセッサのローカルまたは仮想メモリに負担とならないものがあれば有益だろう。さらに、データまたはイベントを待っている間はプロセッサを低電力状態にすることができシステムおよび方法があれば有益だろう。

【課題を解決するための手段】

【0006】

本発明は、プロセッサとメモリ・フロー・コントローラとの間でコマンド・パラメータを通信するためのシステムおよび方法を提供する。本システムおよび方法は、チャンネル

50

・インターフェースを、プロセッサとメモリ・フロー・コントローラとの間の通信のための主要な機構として活用する。チャンネル・インターフェースは、例えば、プロセッサ・ファシリティ、メモリ・フロー制御ファシリティ、マシン・ステータス・レジスタ、および外部プロセッサ割り込みファシリティと通信するためのチャンネルを提供する。これらのチャンネルは、プロセッサのローカル記憶に対する負担を軽減すると共に、バス・トラフィックを最小限にする役割を果たす。

#### 【0007】

これらのチャンネルは、ブロッキングまたは非ブロッキングとして指定されてもよい。ブロッキング・チャンネルによって、対応レジスタから読み出されるべきデータが使用可能でない場合、または対応レジスタに書き込む空間がない場合には、プロセッサは、低電力の「ストール」状態にされる。データが使用可能になるか、または空間が開放されると、プロセッサは、ブロッキング・チャンネル上での通信を介して自動的に呼び起こされる。したがって、本発明のチャンネルにより、プロセッサは、従来技術のシステムにおけるようなイベント・レジスタに対するポーリングまたはスピニングをし続けるのではなく、低電力状態のままでいることができる。

#### 【0008】

本発明のこれらおよび他の特徴および利点は、本発明の例示的な実施形態の以下の詳細な説明に記載されており、または、これを考慮すれば当業者に明らかになるだろう。

#### 【発明を実施するための最良の形態】

#### 【0009】

図1は、本発明の局面が実施されるであろうデータ処理システムのブロック図の例である。図1に例示するデータ処理システムは、インターナショナル・ビジネス・マシン社から入手可能なCELLブロードバンド・エンジン・プロセッサのような、異機種ブロードバンド・プロセッサ・アーキテクチャの実装の例である。ブロードバンド・プロセッサ・アーキテクチャ（BPA）を好適な実施形態の説明に使用するが、本発明はこれに限定されないことは、以下の説明を読めば当業者にとって容易に明らかであろう。

#### 【0010】

図1に示すように、BPA100は、パワー・プロセッサ要素（PPE）110と、複数の相乗作用プロセッサ要素（SPE）120～134との異機種の仕組みを含む。PPE110は、プロセッサ（PPU）116と、そのL1およびL2キャッシュ112および114とを有する。SPE120～134は、それぞれ、自身のプロセッサ・ユニット（SPU）と、メモリ・フロー・コントローラ（MFC）155～162と、ローカルメモリまたは記憶（LS）163～170と、バス・インターフェース・ユニット（BIUユニット）180～194とを有し、BIU180～194は、例えば、直接メモリ・アクセス（DMA）と、メモリ管理ユニット（MMU）と、およびバス・インターフェース・ユニットとの組み合わせであってもよい。高帯域幅内部要素相互接続バス（EIB）196と、バス・インターフェース・コントローラ（MIC）198も提供されている。ブロードバンド・エンジン100は、図1に示す各要素が1つのマイクロプロセッサ・チップに提供されてもよいようなシステム・オン・チップであってもよい。

#### 【0011】

BPA100は、図1に示す各要素が1つのマイクロプロセッサ・チップ上に提供されてもよいようなシステム・オン・チップであってもよい。さらに、BPA100は、各SPUがシステム内の他の各SPUとは異なる命令を受信してもよいような異機種処理環境である。さらに、SPUについての命令セットは、PPUについてのそれとは異なっている。例えば、PPUが縮小命令セットコンピュータ（RISC）ベースの命令を実行し、SPUがベクター化された命令を実行してもよい。

#### 【0012】

SPE120～134は、互いに結合されており、またEIB196を介してL2キャッシュ114に結合されている。加えて、SPE120～134は、EIBを介してMIC198およびBIC197に結合されている。MIC198は、共有メモリ199に対

10

20

30

40

50

する通信インターフェースを提供する。B I C 1 9 7 は、B P A 1 0 0 と、他の外部バスおよび装置との間の通信インターフェースを提供する。

#### 【0013】

P P E 1 1 0 は、二重スレッドの P P E 1 1 0 である。この二重スレッドの P P E 1 1 0 と、8 つの S P E 1 2 0 ~ 1 3 4 との組み合わせによって、B P A 1 0 0 は、1 0 個の同時スレッドと、1 2 8 以上の未処理のメモリ要求とを処理することができる。P P E 1 1 0 は、計算作業負荷の大半を処理する他の 8 つの S P E 1 2 0 ~ 1 3 4 に対するコントローラとしての役割を果たす。例えば、S P E 1 2 0 ~ 1 3 4 がベクトル化された浮動点コードの実行を行う間に、P P E 1 1 0 を使用して従来のオペレーティングシステムを実行してもよい。

10

#### 【0014】

S P E 1 2 0 ~ 1 3 4 は、相乗作用処理ユニット (S P U) 1 4 0 ~ 1 5 4 と、メモリ・フロー制御ユニット 1 5 5 ~ 1 6 2 と、ローカル・メモリまたは記憶 1 6 3 ~ 1 7 4 と、インターフェースユニット 1 8 0 ~ 1 9 4 とを備える。例示的な一実施形態において、ローカル・メモリまたは記憶 1 6 3 ~ 1 7 4 は、P P E 1 1 0 の管理下にあり、ソフトウェアから直接アドレスしてできる 2 5 6 K B の命令およびデータ・メモリを備える。

#### 【0015】

P P E 1 1 0 は、S P E 1 2 0 ~ 1 3 4 に小さなプログラムまたはスレッドをロードして、S P E を互いにつないで複雑な動作における各ステップを処理するようにしてもよい。例えば、B P A 1 0 0 を実装するセット・トップ・ボックスが、DVD、映像および音声復号化、ならびに表示のためのプログラムをロードしてもよく、最終的に出力ディスプレイ上に至るまで、データが S P E から S P E へ渡されることになる。

20

#### 【0016】

メモリ・フロー制御ユニット (M F C) 1 5 5 ~ 1 6 2 は、S P U に対して、残りのシステムおよび他の要素へのインターフェースとしての役割を果たす。M F C 1 5 5 ~ 1 6 2 は、データ転送、保護、およびメイン記憶とローカル記憶 1 6 3 ~ 1 7 4 との間の同期についての主要な機構を提供する。論理的には、プロセッサ内の S P U 毎に M F C がある。実装によっては、複数の S P U 間で 1 つの M F C のリソースを共有できる。そのような場合には、M F C について規定されたすべてのファシリティおよびコマンドは、S P U 毎のソフトウェアに対して独立してみえなければならない。M F C を共有する効果は、実装に依存するファシリティおよびコマンドに限定される。

30

#### 【0017】

##### メモリ・フロー制御 (M F C) ユニット

図 2 は、本発明の例示的な一実施形態に係る典型的なメモリ・フロー制御 (M F C) ユニット 2 0 0 のブロック図の例である。この例示的な実施形態において、M F C 2 0 0 は、S P U に対する 2 つのインターフェース 2 1 0 および 2 1 2 と、バス・インターフェース・ユニット (B I U) 2 2 0 に対する 2 つのインターフェース 2 1 4 および 2 1 6 と、オプションの S L 1 キャッシュ 2 3 0 に対する 2 つのインターフェース 2 2 2 および 2 2 4 とを有する。S P U インターフェース 2 1 0 および 2 1 2 は、S P U チャンネル・インターフェース 2 1 0 および S P U ローカル記憶インターフェース 2 1 2 である。S P U チャンネル・インターフェース 2 1 0 によって、S P U は、M F C ファシリティにアクセスすることと、M F C を発行することができる。S P U ローカル記憶インターフェース 2 1 2 は、S P U 内のローカル記憶にアクセスするために M F C 2 0 0 によって使用される。B I U 2 2 0 に対する一方のインターフェース 2 1 4 によって、M F C ファシリティに対するメモリ・マップ I / O (M M I O) アクセスが可能になる。このインターフェース 2 1 4 によって、他のプロセッサが M F C コマンドを発行することができる。M M I O を使用して発行されたコマンドを、M F C プロキシ・コマンドと称する。

40

#### 【0018】

S L 1 キャッシュに対するインターフェース 2 2 2 および 2 2 4 は、主にデータ転送のためのものである。アドレス変換テーブルに対するアクセスのために M F C 2 0 0 によ

50



て一方のインターフェース222が使用され、メイン記憶とローカル記憶との間のデータ転送のために、他のインターフェース224が使用される。

【0019】

図2に示すように、典型的なMFC内のメイン・ユニットは、メモリ・マップ入力/出力(MMIO)インターフェース240と、MFCレジスタ250と、DMAコントローラ260とを含む。MMIOインターフェース240は、SPUのMFCファシリティをシステムの実アドレス空間にマッピングする。これにより、どのプロセッサまたはシステム内のどの装置からもMFCファシリティにアクセスすることができる。加えて、MMIOインターフェース240は、SPUのローカル記憶を実アドレス空間にマッピングするように構成されてもよい。これにより、どのプロセッサまたはシステム内のどの装置からもローカル記憶にアクセスすることができ、ローカル記憶からローカル記憶への転送と、I/O装置がSPUのローカル記憶ドメインに直接アクセスすることが可能となる。

10

【0020】

MFCレジスタ・ユニット250は、MFCファシリティの大半を含む。あるファシリティは、直接メモリ・アクセス・コントローラ(DMAC)260に含まれる。以下は、MFC200内のファシリティのリストである。ユーザ・モード環境ファシリティ、すなわち、非特権的アプリケーションからアクセスされてもよい環境ファシリティは、(1)メールボックス・ファシリティ、(2)SPU信号通知ファシリティ、(3)プロキシ・タグ・グループ完了ファシリティ、(4)MFCマルチソース同期ファシリティ、(5)SPU制御およびステータス・ファシリティ、および(6)SPU分離ファシリティを含む。特権的モード環境ファシリティ、すなわち、オペレーティング・システムなどの特権的ソフトウェアからのみアクセスされてもよいファシリティは、(1)MFCステート・レジスタ1、(2)MFC論理パーティションIDレジスタ、(3)MFC記憶記述レジスタ、(4)MFCデータ・アドレス・レジスタ、(5)MFCデータ記憶割り込みステータス・レジスタ、(6)MFCアドレス比較制御レジスタ、(7)MFCローカル記憶アドレス比較ファシリティ、(8)MFCコマンド・エラー・レジスタ、(9)MFCデータ記憶割り込みポインタ・レジスタ、(10)MFC制御レジスタ、(11)MFCアトミック・フラッシュ・レジスタ、(12)SPUアウトバウンド割り込みメールボックス・レジスタ、(13)SPU特権的ファシリティ、(14)SPU特権的制御レジスタ、(15)SPUローカル記憶制限レジスタ、(16)SPU構成レジスタ、および(17)SPEコンテキスト退避/復元を含む。

20

30

【0021】

本発明の機構にとって特に重要なファシリティ、すなわち、メールボックス・ファシリティ、SPU信号通知ファシリティ、プロキシ・タグ・グループ完了ファシリティ、MFCマルチソース同期ファシリティ、SPUチャンネル・アクセス・ファシリティ、SPUIイベント・ファシリティ、および割り込みファシリティを、以下により詳細に説明する。

【0022】

データの同期および転送は、一般的には、MFC200内のDMAC260がその役割を担う。DMAC260は、SPUのローカル記憶と、メイン記憶領域との間でデータを移動できる。オプションとして、データはSL1キャッシュにキャッシュできる。

40

【0023】

SPEおよびPPEは、MFC200に対して、コマンド・キュー270および280のうちの1つを介してMFCに対するDMAコマンド要求をキューイングすることによってこれらのDMA動作を行うように命令する。あるSPEによって発行されたコマンドは、MFC・SPUコマンド・キュー280に入れられる。PPEによって発行されたコマンドは、MFCプロキシ・コマンド・キュー270に入れられる。MFCは、メモリ・マッピング・ユニット(MMU)290を使用して、DMA転送に必要な、すべてのMFCアドレス変換およびMFCアクセス保護チェックを行う。

【0024】

MFCコマンドは、SPUにおいて実行する符号がメイン記憶にアクセスしかつシステ

50

ム内の他のプロセッサおよび装置との同期を維持することができる主要な方法を提供する。また、コマンドは、オプションのキャッシュを管理するためにも提供される。MFCコマンドは、SPU上で実行する符号か、またはPPEのような他のプロセッサまたは装置上で実行する符号のいずれかによって発行される。関連SPU上で実行する符号は、MFCコマンドを発行するための一連のチャンネル命令を実行する。他のプロセッサまたは装置上で実行する符号は、MFCコマンドをSPEへ発行するための一連のメモリ・マップI/O(MMIO)転送を実行する。発行されたコマンドは、コマンド・キュー270および280のうちの1つに入れられる。

#### 【0025】

一般的に、コマンドは、MMIOレジスタを使用して、または関連SPUによって実行されたチャンネル命令を通してキューイングすることができる。MMIO方法は、メイン記憶と、関連ローカル記憶との間のデータ転送をSPEに代わってPPEが制御するために使用されるものである。データを転送するMFCコマンドを、MFC・DMAコマンドと称する。MFC・DMAコマンドのためのデータ転送方向は、SPEの観点から常に参照される。したがって、データをSPEへ転送(メイン記憶からローカル記憶へ)するコマンドは、ゲット(get)コマンドとみなされ、データをSPEから転送(ローカル記憶からメイン記憶へ)するコマンドは、プット(put)コマンドとみなされる。

#### 【0026】

MFCコマンドは、コマンド動作に影響を与える複数のパラメータを活用する。図3は、本発明の例示的な実施形態に係るMFCコマンドについてのモニタ・パラメータを示す図の例である。すべてのパラメータがすべてのコマンドによって使用されるわけではない。例えば、EAHパラメータは、オプションである。オプションのパラメータは、コマンド上では指定されないが、オプションのパラメータは、ハードウェアによって「0」に設定される。

#### 【0027】

MFCコマンドは、定義コマンド、違法コマンド、予約コマンドという3つのクラスに分類されてもよい。コマンドのクラスは、操作符号および、もしあれば拡張操作符号を検査することによって決定される。コマンド操作符号、または操作符号および拡張操作符号との組み合わせは、定義または予約コマンドではないので、コマンドは違法である。

#### 【0028】

定義コマンドは、データ転送コマンド、SL1キャッシュ管理コマンド、および同期コマンドという3つのカテゴリーのうちの1つに当てはまる。データ転送コマンドは、データ移動の方向を定義するサブカテゴリーにさらに分割される。プットコマンドは、ローカル記憶からメイン記憶へデータを移動するデータ転送コマンドである。ゲットコマンドは、メイン記憶からローカル記憶へデータを移動するデータ転送コマンドである。アプリケーションは、データ転送コマンドをMFCプロキシ・コマンド・キュー270に置くことができる。特筆されていない限り、これらのコマンドは、任意の順番で(非同期)実行できる。

#### 【0029】

コマンドの「違法」クラスは、定義クラスまたは予約クラスではないコマンドを対象としている。コマンドの「予約」クラスは、実装に依存して使用されているものである。

#### 【0030】

SL1記憶制御コマンドは、SL1キャッシュに関する動作を制御するためのコマンドである。これらの記憶制御コマンドは、例えば、ゲットまたはプット・コマンド、アドレス範囲操作コマンド、フラッシュ・コマンドといった特定の種類のデータ転送コマンドが恐らく発行されることになっていることをSL1キャッシュに対して通知するための「ヒント」コマンドを含む。

#### 【0031】

MFC同期コマンドは、他のMFC、プロセッサ、および装置に関して記憶アクセスが行われる順序を制御するために使用される。MFC同期コマンドは、順序どおりの実行を

実現するためのコマンド、DMAコマンド・キューにおけるバリア・コマンドに先行するすべてのコマンドに関するすべての後続のコマンドを順序付けするためのコマンド、対象信号通知レジスタ内の信号ビットを論理的に設定するための送信信号コマンドなどを含む。

#### 【0032】

MFCコマンドは、個別のDMAコマンドであってもよく、またはDMAリスト・コマンドであってもよい。本発明の例示的な実施形態に係るDMAリスト・コマンドの詳細を図4に示す。DMAリスト・コマンドは、ローカル記憶に記憶された有効アドレスのリスト、転送サイズの対、またはリスト要素を、DMA転送のためのパラメータとして使用する。これらのパラメータは、MFCプロキシ・コマンド・キュー上でサポートされていない、SPUが開始したDMAリスト・コマンドのために使用される。各リスト要素の第1のワードには、転送サイズと、ストール／通知フラグとを含む。第2のワードには、有効アドレスの下位の桁32ビットを含む。開始有効アドレスは、リスト内の転送要素毎に指定される一方で、転送に含まれるローカル記憶アドレスは、主要リスト・コマンドにおいてのみ指定される（「主要(primary)」という用語は、図3に示すパラメータによって指定されるリスト・コマンドのことを指す）。

#### 【0033】

ローカル記憶アドレスは、リストにおける各要素によって転送されるデータ量に基づいて、内部的にインクリメントされる。しかしながら、アラインメントの制約により、ローカル記憶アドレスがリスト要素転送についての16バイト境界で開始しない場合には、ハードウェアは、ローカル記憶アドレスを次の16バイト境界に自動的にインクリメントする。これは、16バイト未満の転送サイズが使用される場合にのみ生じる。16バイト未満の転送サイズのリスト要素は、有効アドレスの最小桁4ビットによって定義されるクワッドワード(16バイト)内のローカル記憶オフセットを使用する。

#### 【0034】

リスト要素内で指定された有効アドレスは、ベースDMAリスト・コマンド内で指定された有効アドレスの上位32ビットによって定義された4GB領域に関連している。DMAリスト開始アドレスは、1つの4GB領域に関連している一方で、リスト要素内の転送は、4GB境界を横断する。

#### 【0035】

「S(ストール／通知)」ビットを設定すると、DMA動作に対して、現在のリスト要素が実行された後にこのリストの実行を中断させ、SPUに対してストール／通知イベントステータスを設定させる。ストールされたリストの実行は、MFCがSPUプログラムからストール／通知確認を受信するまでは再開されない。ストール／通知イベントは、関連コマンド・タグ・グループ・識別子を使用して、SPUプログラムに書き込まれる。ストール／通知要素を有する同一タグ・グループ内に複数のDMAリスト・コマンドがある場合には、ソフトウェアは、タグ別のバリアまたはグローバル・バリアを使用してDMAリスト・コマンドの命令された実行を強制して、曖昧性を避けることを保証する。

#### 【0036】

DMAリスト・コマンド内のすべてのDMAリスト要素は、順番に開始および発行されることが保証される。DMAリスト・コマンド内のすべての要素は、固有のローカル順序を有する。1つのDMAリスト・コマンドは、2048個の要素までを含むことができ、ローカル記憶16KBを占める。

#### 【0037】

##### チャンネル・インターフェース

BPAにおいて、チャンネルは、相乗作用処理ユニット(SPU)とメモリ・フロー制御(MFC)ユニットとの間の主要インターフェースとして使用される。SPUチャンネル・アクセス・ファシリティを使用して、SPUチャンネルを構成、退避、および復元する。SPU命令セット・アーキテクチャ(ISA)は、チャンネル・インターフェース(またはSPUチャンネル)を介した外部装置との通信のためのチャンネル命令の組を提供

10

20

30

40

50

する。表1は、これらの命令を列記している。

【0038】

【表1】

チャンネル命令

チャンネル命令	二一モニック命令	動作説明
チャンネル読み出し	rdch	アドレス指定チャンネルに記憶されたデータの読み出しを、選択された汎用レジスタ (GPR) にロードさせる
チャンネル書き込み	wrch	選択されたGPRからデータを読み出し、アドレス指定チャンネルに記憶させる
チャンネルカウント読み出し	rchcnt	アドレス指定チャンネルに関連したカウントを、選択されたGPRに記憶させる

10

【0039】

アーキテクチャ的には、SPUチャンネルは、読み出し専用または書き込み専用のアクセス種類を有するように構成される。チャンネルは、読み出しおよび書き込みとして構成することはできない。アクセス種類に加えて、各チャンネルは、非ブロッキングまたはブロッキングとして構成できる。ブロッキングとして構成されたチャンネルは、SPUに対して、チャンネル・カウント「0」のチャンネルを読み出す場合またはフル・チャンネル（すなわち、チャンネル・カウント「0」のチャンネル）に書き込む場合にはストールさせる。「読み出し」チャンネルとは、チャンネル読み出し命令 (rdch) のみがこのチャンネルに発行でき、データが常に返されることを意味する。「書き込み」チャンネルとは、チャンネル書き込み命令 (wrch) のみがこのチャンネルに発行でき、データは当該チャンネルに常に受理される。

20

【0040】

「読み出しブロッキング」チャンネルとは、チャンネル読み出し命令 (rdch) のみがこのチャンネルに発行できることを意味する。読み出しブロッキング・チャンネルへ送られたチャンネル読み出し命令 (rdch) は、チャンネル・カウントがゼロでない場合にのみ完了する。チャンネル・カウント「0」は、チャンネルが空であることを示す。チャンネル読み出し (rdch) を読み出しブロッキング・チャンネルに対してカウント「0」で実行すると、データが当該チャンネル内で使用可能になるまでSPUストールとなる。

30

【0041】

「書き込みブロッキング」チャンネルとは、チャンネル書き込み命令 (wrch) のみがこのチャンネルに発行できることを意味する。書き込みブロッキング・チャンネルへ送られたチャンネル書き込み (wrch) 命令は、チャンネル・カウントがゼロでない場合にのみ完了する。チャンネル・カウント「0」は、チャンネルがフルであることを示す。チャンネル書き込み (rdch) 命令を書き込みブロッキング・チャンネルに対してカウント「0」で実行すると、アドレス指定されたチャンネル内のエントリが使用可能になるまでSPUストールとなる。

40

【0042】

チャンネル構成に不適切なチャンネル命令を発行すると、無効なチャンネル命令割り込みという結果になることに注意すべきである。例えば、チャンネル読み出し命令 (rdch) を書き込みまたは書き込みブロッキング・チャンネルに対して発行すると、無効なチャンネル命令割り込みという結果になる。

【0043】

50

各チャンネルは、当該チャンネルについて発行可能な未処理の動作の数を示す対応カウント（すなわち、深度）を有する。チャンネル深度（すなわち、未処理の転送の最大数）は、実施に依存する。ソフトウェアは、SPU内で新たなコンテキストを確立する場合、または既存のコンテキストを再開させる場合には、チャンネル・カウントを初期化しなければならない。

#### 【0044】

チャンネルおよびチャンネル・インターフェースの動作は、2003年6月26に出願の同時係属および本発明の譲受人に譲渡された米国特許出願公開第2004/0264445号、「External Message Passing Method and Apparatus」に記載されており、その内容を参照により本願に引用するものと10する。図5は、米国特許出願公開第2004/0264445号に記載された機構に係るチャンネルの単一の対のためのチャンネル回路に関するSPU発行／制御論理の仕組みならびにデータ・フローを示す図の例である。チャンネル・インターフェースの動作を、図5に示す様々なブロックを参照して説明する。

#### 【0045】

図5に示すように、ブロック430は、SPUの外部装置命令発行／制御論理である。ブロック432は、SPUへおよびそこからのデータ・フローを表わす。周知のように、プロセッサは、数多くの互いに異なる外部装置と同時に通信してもよい。このプロセッサにおいて、通信はチャンネル・レジスタを介して達成される。各チャンネルは、1方向にのみ動作し、チャンネル上においてSPUによって行われうる動作に従って、読み出しチャンネルまたは書き込みチャンネルのいずれかと呼ばれる。ブロック434は、ブロック435によって表わされるような1つの外部装置のためのチャンネルの1組についてのチャンネル論理を表わす。後でさらに詳細に説明するように、この外部装置435は、MFC200、マシン・ステータス・レジスタ、または任意の他の種類の外部装置のようなMFCであってもよい。特に、MFC、マシン・ステータス・レジスタ、イベント・ファシリティ、メールボックス・ファシリティ、および信号通知ファシリティと通信するためのチャンネル・インターフェースの使用については、チャンネル・インターフェースについての以下のこの一般的な説明において説明する。

#### 【0046】

ブロック434内部において、チャンネル読み出しカウンタ436と、読み出しレジスタ436と、チャンネル書き込みカウンタ440と、書き込みレジスタ442と、MUX（マルチプレクサ）444と、MUX446とが示されている。チャンネル命令は、バス448上で、SPU発行／制御論理430から読み出し及び書き込みカウンタ436および440へ送付されると共に、MUX444および336のゲート入力へも送付される。これらの命令は、チャンネルOUTリード450上で、435のような適切な外部装置に対しても供給される。データINリード452は、外部装置435から読み出しレジスタ438へデータを提供する。チャンネル・カウントIN信号は、チャンネルINリード454上で、外部装置435から、データがレジスタに入力されたことを示してカウンタ436内のカウントを値または桁1つ分変更するように動作するカウンタ436へ供給される。

#### 【0047】

書き込みレジスタ442から外部装置435へ出力されているデータは、データOUTリード上で提供される。チャンネル確認信号は、外部装置435が満足のいくデータ受信を完了してカウンタ440内のカウントを値または桁1つ分変更するように動作する場合に、チャンネルACKリード458上で、外部装置435から書き込みチャンネル・カウンタ440へ返される。本発明の好ましい一実施形態において、バス448上の信号は、適切な読み出しまたは書き込みカウンタをデクリメントする一方で、リード454または458のいずれかの上の信号は、適切な読み出しまたは書き込みカウンタをインクリメントすることになる。

#### 【0048】

10

20

30

40

50

図示のように、カウンタ436および440の両方のカウントは、MUX444を通して、チャンネル・ストール・リード460上で、SPU発行／制御論理430へ供給される。チャンネル書き込みデータは、SPUデータ・フロー・ブロック432から、チャンネル書き込みデータ・リード462上で、書き込みレジスタ442へ供給される。ブロック436、438および440からの出力は、バス464上でデータ・フロー・ブロック432へ返される。非チャンネル命令は、バス466を介して、ブロック430および432間で通信される。

#### 【0049】

図6は、本発明の例示的な一実施形態に係るチャンネル・インターフェースの動作の例の概略を示すフローチャートを提供する。図6におけるフローチャート図の各ブロックおよびブロックの組み合わせ、および以下に説明する後続の図面におけるフローチャート図は、コンピュータ・プログラム命令によって実施することができることが理解されるだろう。これらのコンピュータ・プログラム命令は、プロセッサまたは他のプログラム可能なデータ処理装置へ提供されて、プロセッサまたは他のプログラム可能なデータ処理装置上で実行する命令がフローチャートのブロックまたは複数のブロックにおいて指定された機能を指示するための手段を作成するようなマシンを生じさせてもよい。これらのコンピュータ・プログラム命令は、プロセッサまたは他のプログラム可能なデータ処理装置が特定のやり方で機能するように指示することができるコンピュータ読み出し可能なメモリまたは記憶媒体に記憶されてもよい。特定のやり方とは、コンピュータ読み出し可能なメモリまたは記憶媒体に記憶された命令が、フローチャートのブロックまたは複数のブロックにおいて指定された機能を実施する指示手段を含む製品を生じさせるようなやり方である。

#### 【0050】

従って、フローチャート図のブロックは、特定の機能を実行するための手段の組み合わせ、特定の機能を実行するためのステップの組み合わせ、および当該特定の機能を実行するためのプログラム命令手段をサポートする。また、フローチャート図の各ブロックおよびフローチャート図のブロックの組み合わせは、特定の機能またはステップを実行する専用のハードウェア・ベースのコンピュータ・システムによって、または専用のハードウェアおよびコンピュータ命令の組み合わせによって実施できることが理解されるだろう。

#### 【0051】

図6に示すように、チャンネル読み出しまたは書き込みコマンドの発行によって、指定のチャンネルが上述のような制御機構が実装されているものであるかどうかを判断することになる(ステップ576)。そうなければ、チャンネル・エラー論理がイネーブルかどうかについて決定する(ステップ578)。イネーブルであれば、プロセッサは停止する(ステップ580)。イネーブルでなければ、コマンドが読み出しか書き込みかどうかの判断がなされる(ステップ582)。

#### 【0052】

非実装コマンドが書き込みである場合は、当該コマンドに対してはこれ以上何も行われない(ステップ584)。それに対して、非実装コマンドが読み出しである場合には、データ・プロセッサ・データ・フローへゼロを返す(ステップ586)。いずれの場合も、処理は次の読み出しまたは書き込み命令を待つステータスへ戻る。図示の好ましい実施形態において、すべての有効な読み出し命令は、値を返さなければならない。本明細書に規定するように、非実装チャンネルのチャンネル読み出し命令は、すべてゼロの値を返す。

#### 【0053】

なお、特定の実施においては、チャンネルのすべてを定義する必要はない。各チャンネルは、固有の数値識別子を有することになる。好ましい一実施形態において、このチャンネル識別子は、0から127までの範囲である。しかしながら、チャンネルのすべてを定義する必要はないので、識別子のすべてが使用されなくてもよい。よって、未定義チャンネルへの命令がある場合には、処理は、上述の非実装経路へと下る。実施によっては、非実装チャンネルへのチャンネル読み出しまたは書き込みコマンドは、違法動作であるとみなされるのが望ましい場合がある。前述のステップ580に示すように、さらなる行為に

10

20

30

40

50

よってプロセッサが強制的に停止されることもある。

【0054】

図6に戻ると、ステップ576において、指定されたチャンネルが実装されていると判断される場合には、指定されたチャンネルがブロッキング・チャンネルかどうかをチェックする（ステップ588）。そうでなければ、当該チャンネルについてのカウントがデクリメントされるが、0未満になることは許可されていない（ステップ590）。チャンネルがブロッキングであると判断される場合には、当該チャンネルについてのカウントがゼロより大きいかどうかを判断するためのチェックが行われる（ステップ592）。ゼロより大きければ、処理はステップ590へ戻る。

【0055】

カウントが既にゼロである場合には、ステップ592において判断されるように、SPUは、外部装置がこのチャンネルに関する入力を提供してカウントをゼロから変更させるまで、ストールする（ステップ594および595）。よって、ステップ594および595のループは、このチャンネルについてのカウントに変化があるまで、定期的に処理される。一旦カウントが変更されると、処理はステップ595からステップ590へと続く。

【0056】

その後、チャンネルが能動か受動かを判断する（ステップ596）。チャンネルが受動である場合には、コマンドが書き込みまたは読み出し命令かどうかをチェックする（ステップ598）。コマンドが書き込み命令である場合には、データは外部読み出しのためのローカルに記憶される（ステップ600）。コマンドが読み出し命令である場合には、データは、図5のSPUデータ・フロー432を介してSPUへ返される（ステップ602）。

【0057】

受動チャンネルである場合に、SPUは、処理を完了するのに外部処理に依存することに注意すべきである。一例として、読み出しチャンネルは、データをロードするのに外部装置に依存してもよい。これに対して、能動チャンネルにおいて、SPUは、読み出しまたは書き込み動作を実行する動作を能動的に完了させる。この種の動作の一例として、接続されたハードウェアが能動読み出しチャンネルからデータに対する外部要求を行う場合が挙げられる。

【0058】

チャンネルが能動チャンネルであるとステップ596において判断されると、コマンドが読み出しまたは書き込みコマンドかどうかをチェックする（ステップ604）。コマンドが書き込みのためのものである場合は、書き込みデータを、SPUへ外部に、または内部レジスタに出力する（ステップ606）。コマンドが読み出しである場合には、読み出し要求を適切な外部装置へ送る（ステップ608）。

【0059】

要求されたデータの入力を待つ（ステップ610）。定期的に、読み出しデータが受信されたかどうかについて判断する（ステップ612）。そうでなければ、処理は、次のチェックのための時間が生じるまで、ステップ610へ戻る。データが受信されると、処理は完了する（ステップ602）。

【0060】

上記から、チャンネル番号がその内部で指定される、特定のチャンネル読み出しまたは書き込み命令を使用して、各チャンネルがアクセスされることは明らかであろう。各チャンネルは、それによって指定されたカウントを有する。このカウントは、当該チャンネルがその内部で指定される、チャンネル読み出しカウント命令を使用して、読み出される。チャンネルコマンドは、投機的ではなく、外部のインターフェースにおいて順序正しくなく処理されることはできない。チャンネル・アーキテクチャは、SPU外部の装置がチャンネルコマンドを順序通りに処理することを要求してはいないが、プロセッサおよび外部装置の実施によっては、それを要求することもある。このカウントレジスタ内の値は、こ

10

20

30

40

50

のレジスタへのアクセス数を、このレジスタに生じた外部確認の数と対比して常に追跡している。

#### 【0061】

動作において、チャンネル・カウンタを（複数の）外部インターフェースを通じたアクセスを介して変更するやり方は、実施に基づいている。好ましい一実施形態において、カウンタは、レジスタへまたはそこからの成功したデータ転送毎に1ずつインクリメントされる。チャンネル毎に、S P Uアクセスは、読み出しまたは書き込みチャンネルとして定義される。さらに、好ましい実施形態において、「0」カウンタを使用して、チャンネルが「ブロッキング」チャンネルとして定義または実施される場合にこれ以上の動作をストールする。チャンネル・レジスタがキュー深度「1」を有すると定義される場合には、「0」カウンタを使用して、当該データ内のデータが有効でない旨を示してもよい。チャンネルは、読み出しまたは書き込みチャンネル上でのS P U動作をストールするために定義することもでき、当該コマンドにおいて、カウンタがもはや「0」でないようなときになるまでカウンタが「0」の場合であれば、このようにできる。

10

#### 【0062】

好ましい一実施形態において、カウンタ値は、S P Uが開始したチャンネル読み出しまたは書き込みコマンド毎にデクリメントされ、外部が開始した読み出しまたは書き込み（データありまたはデータなし）アクセス毎にインクリメントされる。言い換えれば、カウンタは、入力対出力を示すことを維持する。よって、値またはカウンタ「0」は、書き込みに関しては、外部書き込みスロットがこれ以上使用可能ではないことを示す。それに対して、読み出しに関する値またはカウンタ「0」は、有効なデータがないことを示す。カウンタがゼロの場合に、追加のS P Uチャンネル読み出しまたは書き込みコマンドが発行されてチャンネルが非ブロッキングと定義される場合には、カウンタは「0」のままとなり、レジスタ内のデータは失われる。好ましい実施形態において実施されるように、当該レジスタの以前の直近データが失われる。カウンタが当該レジスタの実施のビット数の最大値にあって、さらにトランザクションがあるとカウンタに範囲外のインクリメントを生じさせる場合には、カウンタは最大値に留まる。

20

#### 【0063】

カウンタ値を初期化する方法は実施に依存し、その方法の1つは、外部インターフェースを介する初期化である。このカウンタは、外部キューの深度に予め設定できる。カウンタレジスタ内のゼロの値は、この外部キュー内にこれ以上空間がないことを意味する。外部キュー深度「1」については、カウンタは「1」に予め設定されなければならない。S P Uがこのチャンネルに書き込む場合、カウンタは「0」となる。外部装置がこのチャンネルを読み出す場合、カウンタは「1」にインクリメントされ、それによって、チャンネルが他の書き込み動作に対して準備ができていることを示す。

30

#### 【0064】

上述のように、チャンネルレジスタの読み出しによって、カウンタは有効データを示すことができる。カウンタレジスタが「0」に予め設定されると、これはデータが有効でないことを示す。外部装置がこのチャンネルに書き込みを行う場合には、カウンタは「1」にインクリメントし、データはS P U読み出しについて有効である旨を示す。S P Uがこのチャンネルから読み出しを行う場合には、カウンタは「0」にデクリメントし戻されて、他の外部書き込みが生じうることを示す。

40

#### 【0065】

本発明の好ましい一実施形態において、コンピュータ符号チャンネル・カウンタ読み出し命令がカウンタへ送られて、読み出しおよび書き込みチャンネル両方に関するカウンタを確認する。外部装置がマルチプロセッサ環境における他のS P Uまたはコンピューティング装置のようなインテリジェント装置である場合は、外部装置は、チャンネル・カウンタ読み出し命令をカウンタに送って、カウンタを確認してもよい。このように、外部装置は、チャンネルが読み出しまたは書き込みチャンネルのいずれかに読み出しされていないデータを含む時、または、読み出しチャンネルを含むプロセッサへ追加のデータを送付す

50



るのが適切な時、あるいはその両方を判断してもよい。

【0066】

本発明の使用において、読み出しおよび書き込みチャンネルは、非累積または累積のいずれであってもよい。累積チャンネルは、複数の書き込みを累積するチャンネルである。すなわち、チャンネルが読み出されるまで、レジスタまたは他の記憶手段に既に含まれるデータに入力データが論理的に追加される。チャンネルを読み出す場合に、累積レジスタは、典型的には「0」に再設定されて、チャンネルは、再び累積を開始する。この行為は、読み出しまたは書き込みチャンネルの両方について可能である。

【0067】

さらに、累積チャンネルは、ブロッキングであっても、または非ブロッキングであってもよい。典型的には、累積チャンネルはカウント深度「1」を有するのに対して、非累積チャンネルは、当該チャンネルへの書き込み毎にカウントを行ってもよい。

【0068】

要約すると、チャンネル・インターフェースは、定義されたチャンネルを使用してメモリを開放するが、それでも、レジスタ内のデータがいつ有効になるか、または、言い換えれば、かつて読み出されていないときについてのアクセス可能な情報を容易に提供する。この情報は、チャンネル・カウント読み出し命令をカウント機構へ送ることによって主翼される。インテリジェント外部装置が所定のチャンネルに接続されている場合には、所定のチャンネルへデータを送信、またはそこからデータ受信する際に、同様の命令が外部装置によって使用されてもよい。チャンネル・インターフェースは、チャンネル・カウント読み出し命令を使用することによって、指定されたチャンネルが非ブロッキング・チャンネルとして定義される場合に、レジスタ内のデータの不測の上書きをさらに防止する。

【0069】

本発明は、チャンネル・インターフェースおよび定義されたチャンネルを利用して、互いに異なる種類の外部装置およびそのような外部装置によって提供されたファシリティへ、またはそこから、命令およびデータを通信する。例えば、本発明は、SPUのチャンネル・インターフェースのを使用して、命令およびデータを、MFC、マシン・ステータス・レジスタ、および割り込みファシリティと通信するための機構を提供する。さらに、チャンネル・インターフェースを使用して、命令およびデータを、BPAのイベント・ファシリティ、メールボックス・ファシリティ、マルチソース同期ファシリティ、プロキシ・タグ・グループ完了ファシリティ、信号通知ファシリティなどと通信する。

【0070】

図7は、チャンネルが本発明の一実施形態によって使用されるやり方を示す図の例である。図7に示すように、チャンネルインターフェース620は、複数のチャンネルを提供し、それを介して、MFC624、マシン・ステータス・レジスタ634、および割り込みファシリティ636と通信してもよい。各チャンネルは、図5に関連して上述したものと同様の要素からなってもよく、その動作は、図6において説明したものと同様である。本発明の一実施形態において、チャンネル・インターフェース620は、図5におけるブロック434によって表わされるすべてのチャンネル対の総計に対応してもよい。SPU622は、図5におけるブロック430および432の組み合わせに対応してもよく、図7~11におけるすべての他のブロックは図5のブロック435に対応してもよい。

【0071】

図7に示すように、チャンネル631、633、および635は、SPUイベント・ファシリティ630とデクリメンタ632とがMFC624と通信できるようなSPU622に関する通信経路を提供する。SPUイベント・ファシリティ630は、BPA内で生成されたイベントを処理するための機構を提供する。チャンネル633は、後にさらに少佐に説明するように、当該イベントを識別して、当該イベントに関する情報を取得するための機構を提供する。デクリメンタ632は、SPU上で実行するソフトウェアが時間の進行を計測でき、または所定の時間単位の経過が通知される機構を提供する。デクリメンタ632は、その値セットおよびチャンネル631を介して読み出されたステータスを有

してもよい。

【0072】

S P U（発行／制御論理、プロセッサ・データ・フロー）622は、外部装置と通信するための命令、データおよびファシリティを提供する。例えば、S P U 622は、S P Uチャンネルを初期化、退避、および復元する特権的ファシリティであるS P Uチャンネル・アクセス・ファシリティを提供する。当該ファシリティは、S P Uチャンネル・インデックス・レジスタ、S P Uチャンネル・カウント・レジスタ、およびS P Uチャンネル・データ・レジスタという3つのM M I Oレジスタからなる。P Uチャンネル・インデックス・レジスタは、S P Uチャンネル・カウント・レジスタおよびS P Uチャンネル・データ・レジスタによってアクセスされたそれぞれカウントおよびデータのチャンネルへのポインタである。S P Uチャンネル・インデックス・レジスタは、S P Uチャンネル・カウント・レジスタまたはS P Uチャンネル・データ・レジスタを使用してアクセスされたS P Uチャンネルを選択する。読み出したまたはS P Uチャンネル・インデックス・レジスタによって選択されたS P Uチャンネル・データを初期化するために、S P Uチャンネル・データ・レジスタが使用される。

10

【0073】

S P U 622のファシリティと通信するためのチャンネル631、633、および635に加えて、チャンネル637は、マシン・ステータス・レジスタ634に関連した通信経路を提供する。マシン・ステータス・レジスタ634は、現在のマシン・分離ステータスと、割り込みステータスとを含む。分離ステータスは、S P Uが分離されているか否かを示す。B P Aの分離ファシリティによって、特権的ソフトウェアおよびアプリケーションは、1つ以上のS P Uへ符号画像をロードすることができる。S P U分離ファシリティは、S P Uの関連ローカル記憶にロードされた符号画像が決して変更されなかった旨を保証する。入れ子型の割り込みがサポートされる場合には、割り込みステータス関連のマシン・ステータス・レジスタを使用して割り込み状態情報を退避する。

20

【0074】

さらに、チャンネル639は、割り込みファシリティ636に関連した通信経路を提供する。割り込みファシリティ636は、割り込みおよび割り込みステータス情報をP P Eまたは外部装置にルーティングし、P P Eに提供される割り込みを優先させて、内部プロセッサ割り込みを生成するのに使用される。

30

【0075】

加えて、チャンネル641は、メールボックス・ファシリティ638に関連した通信経路を提供する。他のS P U、P P Eなどのような外部装置へ、およびそこから情報を送受信するために、メールボックス・ファシリティ638が使用される。

【0076】

チャンネル643は、S P U信号通知ファシリティ640に関連した通信経路を提供する。バッファ完了フラグのような信号をシステム内の他のプロセッサおよび装置からS P Uへ送るために、S P U信号通知ファシリティ640が使用される。

【0077】

チャンネル645は、プロキシ・タグ・グループ完了ファシリティ642に関連した通信経路を提供する。プロキシ・タグ・グループ完了ファシリティ642は、同様にタグ付けされた命令のグループの処理が完了したときを判断するためのファシリティである。

40

【0078】

チャンネル647は、M F Cマルチソース同期ファシリティ644に関連した通信経路を提供する。M F Cマルチソース同期ファシリティ644は、ローカル記憶およびメイン記憶アドレス・ドメインに渡る累積順序付けを達成する。複数のソース（すなわち、2つ以上のプロセッサまたはユニット）によって他のプロセッサまたはユニットに対して行われる記憶アクセスの順序付けを、累積順序付けと称する。

【0079】

図8および図9は、本発明の例示的な一実施形態に係るS P Uチャンネル・マップを表

50

で示す図の例である。図7～図11に示すように、SPUチャンネル・インターフェースは、命令およびデータを通信するための様々な種類のチャンネルをサポートしている。これらのチャンネルには、SPUイベント・チャンネル650と、SPU信号通知チャンネル652と、SPUデクリメンタ・チャンネル654と、MFCマルチソース同期チャンネル656と、SPU予約チャンネル658と、マスク読み出しチャンネル660と、SPU状態管理チャンネル662と、MFCコマンド・パラメータ・チャンネル664と、MFCタグ・ステータス・チャンネル666と、SPUメールボックス・チャンネル668とが含まれる。これらの「チャンネル」は、実質的には、メモリ・マップ・レジスタおよびこれらのレジスタに書き込むための対応回路である。よって、「チャンネル」という用語は、本明細書においては、指定された「チャンネル」に対応するデータ値を記憶するための単数または複数のレジスタを指すためにも使用されてもよい。これらの各チャンネルの動作は、後述する。これらのチャンネルを使用するためのPPE、SPU、およびMFUにおいて、様々なファシリティが提供される。これらの種類の各チャンネルを、MFCと通信するために使用されるチャンネルから始めて、以下に詳細に説明する。

#### 【0080】

##### MFCコマンド・パラメータ・チャンネル

MFCコマンド・パラメータ・チャンネル664は、MFC・SPUコマンド・キューのコマンド・パラメータ・レジスタヘデータを書き込むために使用されるチャンネルである（図2および上記表1を参照）、MFCコマンド・パラメータ・チャンネル664は、非ブロッキングであり、関連のチャンネル・カウントを有しない。よって、これらのチャンネルのいずれに送られるチャンネル・カウント読み出し（rchcnt）命令も、カウント「1」を返す。

#### 【0081】

MFCコマンド・パラメータ・チャンネル664は、MFCローカル記憶アドレス・チャンネルと、MFC有効アドレスhighチャンネルと、MFC有効アドレスlowまたはリスト・アドレス・チャンネルと、MFC転送サイズまたはリスト・サイズ・チャンネルと、MFCコマンド・タグ識別チャンネルと、MFCコマンド操作符号チャンネルと、MFCクラスIDチャンネルとを含む。これらの各チャンネルを、以下に詳細に説明する。

#### 【0082】

##### MFCコマンド操作符号チャンネル

MFCコマンド操作符号チャンネルの詳細を、本発明の例示的な一実施形態に係る図12に示す。MFCコマンド操作符号チャンネルは、操作符号に基づいて実行されるべき動作を識別する。この操作符号の有効性は、命令ストリームとは非同期でチェックされる。MFCコマンドまたは任意のコマンドパラメータが無効である場合には、MFCコマンド・キュー処理は中断されて、無効なMFCコマンド割り込みが生成される。

#### 【0083】

MFCコマンドおよびクラスIDパラメータは、1つのチャンネル命令を使用して、MFC・SPUコマンド・キューへ書き込まれる。図12に示すように、一実施形態において、MFCコマンド操作符号パラメータは、32ビットワードの下位16ビットである。このフィールドの上位8ビットは、MFCコマンド操作符号を識別する。

#### 【0084】

##### MFCクラスIDチャンネル

MFCクラスIDチャンネルは、図13に示すように、MFCコマンド毎の置換クラスIDおよび及び転送クラスIDを指定するために使用される。これらのIDは、SPUおよびソフトウェアによって使用されて、システムの全体的な性能を向上させる。特に、置換クラスID（RclassID）は、置換管理テーブル（RMT）と共に使用されて、キャッシュ置換を制御する。置換クラスIDは、例えば、PPE動作についてのロード・記憶アドレスから生成されてもよい（PPEは、PPEのロード・記憶ならびにキャッシュ置換管理ファシリティのためのクラスIDへの命令フェッチについての有効アドレスま

10

20

30

40

50

たは実アドレスをマッピングする方法を提供するアドレス範囲ファシリティを含む)。

【0085】

RclassIDは、特権的ソフトウェアによって管理されたテーブル、すなわち、置換管理テーブル(RMT)へのインデックスを生成するために使用される。RMTは、置換ポリシーを制御するために使用される。RMTのフォーマットは、実施に依存する。RMTは、実施に依存するエントリ数からなり、セット・イネーブルビットと、有効ビットと、他の制御情報を含まなければならない。オプションとして、ある実施においては、キャッシュ・バイパス・ビットおよびアルゴリズム・ビットが提供される。RMT内のエントリの数および各エントリのサイズは、実施に依存する。

【0086】

図14は、本発明の例示的な一実施形態に係る8ウェイ・セット・アソシエティブ・キャッシュに対する典型的なRMTエントリを示す。RMTテーブルは、システムの実アドレス空間に位置する。特権的ソフトウェアは、これらのRMTテーブルを特権的ページとしてマッピングしなければならない。ある実施においては、主要なキャッシュ構成毎にRMTが提供されなければならない。

【0087】

図13に戻ると、転送クラスID(TclassID)は、互いに異なる特徴を有する記憶へのアクセスを識別するために使用される。TclassIDを使用して、ある実施においては、記憶場所の特徴に基づいて、MFCコマンドに対応する転送を最適化することができる。TclassIDのセットアップおよび使用は、実施に依存する。

【0088】

RclassIDおよびTclassIDは、以後「クラスIDパラメータ」と称し、その内容は永続的ではなく、コマンド・エンキュー・シーケンス毎に書き込まなければならない。クラスIDパラメータは、共に使用するコマンドがPPEから発行されたか、それともSPUコマンド・キューのSPU側から発行されたかに関わらず、同一の機能を実行する。クラスIDパラメータは、SPEに関連したリソースを制御するために使用され、他のSPEまたはPPEに関連したリソースへは影響を与えない。クラスIDパラメータの有効性は、検証されない。サポートされるクラスIDパラメータの数は、実施に依存する。

【0089】

MFCコマンド・タグ命令チャンネル

MFCコマンド・タグ命令チャンネルは、コマンド毎、またはコマンドのグループ毎に識別子を指定するために使用される。MFCコマンド・タグ識別チャンネルの詳細を、図15に示す。識別タグは、例えば、x'0'とx'1F'との間の任意の値である。識別タグは、ハードウェアにおける単なるローカルな範囲を有する。よって、同一のタグを互いに異なるSPEまたはPPEにおいて使用してもよい。

【0090】

任意の数のMFCコマンドは、同一の識別子でタグ付けすることができる。同一の識別子でタグ付けされたMFCコマンドは、タグ・グループと称される。タグは、特定のキューへ書き込まれたコマンドに関連する。MFC・SPUコマンド・キューへ供給されたタグは、MFCプロキシ・コマンド・キューへ供給されたタグとは独立している。MFCコマンド・タグ識別パラメータは、永続的ではなく、MFCコマンド・エンキュー・シーケンス毎に書き込まなければならない。このパラメータの有効性は、命令ストリームとは非同期でチェックされる。上位ビット(例えば、ビット0から10)が0に設定されていなければ、MFCコマンド・キュー処理は中断されて、割り込みが生成される。

【0091】

MFC転送サイズまたはリスト・サイズ・チャンネル

MFC転送サイズまたはリスト・サイズ・チャンネルは、MFC転送のサイズ、またはMFC・DMA転送リスト、すなわち、一連のDMA転送コマンドのリストのサイズを指定するために使用される。MFC転送サイズまたはリスト・サイズ・チャンネルの詳細を

10

20

30

40

50

図16に示す。例示的な一実施形態において、転送サイズは、1, 2, 4, 8, 16, または16バイトから最大16KBまでの倍数を有することができる。MFC・DMA転送リスト・サイズは、8という値か、または最大16KBまでの8の倍数かを有することができる。MFC転送サイズまたはリスト・サイズ・チャンネルの内容は、永続的ではなく、MFCコマンド・エンキュー・シーケンス毎に書き込まなければならない。このパラメータの有効性は、命令ストリームとは非同期でチェックされる。サイズが無効であれば、MFCコマンド・キュー処理は中断されて、MFC・DMAアラインメント割り込みが生成される。

#### 【0092】

##### MFCローカル記憶アドレス・チャンネル

MFCローカル記憶アドレス・チャンネルは、キューイングされたMFCコマンドに関連したSPUローカル記憶アドレスを供給するために使用される。MFCローカル記憶アドレスは、MFCコマンド内で定義されるように、MFC転送の送信元または送信先として使用される。MFCローカル記憶アドレス・チャンネルの詳細を図17に示す。

#### 【0093】

MFCローカル記憶アドレス・チャンネルの内容は、永続的ではなく、MFCコマンド・エンキュー・シーケンス毎に書き込まなければならない。MFCローカル記憶アドレス・パラメータの有効性は、命令ストリームとは非同期でチェックされる。アドレスが整列していない場合には、MFCコマンド・キュー処理は中断されて、MFC・DMAアラインメント例外が生成される。整列されているとみなされるためには、例えば、ローカル記憶アドレスの最小桁4ビットが、有効アドレスの最小桁4ビットに一致しなければならない。

#### 【0094】

##### MFC有効アドレスLowまたはリスト・アドレス・チャンネル

MFC有効アドレスLowまたはリスト・アドレス・チャンネルは、MFCコマンドについての有効なLowアドレス、またはMFC・DMAリスト・コマンドについてのリスト要素へのローカル記憶ポインタを指定するために使用される。MFCステートレジスタ内で変換がイネーブルな場合には、PPEのアドレス変換ファシリティによって有効アドレスが実アドレスに変換される。図18は、MFC有効アドレスLowまたはリスト・アドレス・チャンネルの詳細を示す。

#### 【0095】

MFC有効アドレスLowまたはリスト・アドレス・チャンネルの内容は、永続的ではなく、MFCコマンド・エンキュー・シーケンス毎に書き込まなければならない。16バイト未満の転送サイズについては、このパラメータのビット28から31が転送サイズに基づく自然整列を提供しなければならない。16バイト以上の転送サイズについては、ビット28から31が「0」でなければならない。変換がディスエーブルの場合は、このパラメータは、メイン記憶ドメインの実アドレス空間制限内でなければならない。MFCリスト・コマンドについては、リスト・アドレスのビット29から31が「0」でなければならない。これらの条件のいずれかが満たされない場合には、パラメータは無効であり、整列されていないとみなされる。

#### 【0096】

MFC有効アドレスLowまたはリスト・アドレス・パラメータの有効性は、命令ストリームとは非同期でチェックされる。例えばセグメント障害、マッピング障害、保護違反、またはアドレスが整列されていないのでアドレスが無効の場合は、MFCコマンド・キュー処理は中断されて、割り込みが生成される。生成される可能性のあるこの種類の割り込みは、MFCデータ・セグメント割り込み、MFCデータ記憶割り込み、およびDMAアラインメント割り込みである。

#### 【0097】

##### MFC有効アドレスHighチャンネル

MFC有効アドレスHighチャンネルは、MFCコマンドについての有効アドレスを

10

20

30

40

50

指定するために使用される。変換がMFC状態レジスタ内でイネーブルの場合は、有効アドレスは、アドレス変換ファシリティによって実アドレスに変換される。MFC有効アドレスhighチャンネルの詳細を図19に示す。

#### 【0098】

MFC有効アドレスhighチャンネルの内容は、永続的ではなく、MFCコマンド・エンキュー・シーケンス毎に書き込まなければならない。上位32ビットが書き込まれていない場合には、ハードウェアはEAHを設定し、高アドレスビットが0、すなわちアドレスが0から4GBであるように設定される。このパラメータの有効性は、命令ストリームとは非同期でチェックされる。例えばセグメント障害、マッピング障害、または保護違反のためにアドレスが無効の場合は、MFCコマンド・キュー処理は中断されて、割り込みが生成される。生成される可能性のあるこの種類の割り込みは、MFCデータ・セグメント割り込み、およびMFCデータ記憶割り込みである。有効アドレスの有効性は、転送中にチェックされることに注意すべきである。無効アドレスとなって例外が生成される前であれば、部分的な転送が可能である。

10

#### 【0099】

上述のMFCコマンド・パラメータ・チャンネルを使用して、SPUからMFCコマンドをキューイングするためには、MFCコマンド・パラメータは、まずMFCコマンド・パラメータ・チャンネルに書き込まなければならない。これは任意の順序で行われてもよいが、MFCコマンド操作符号およびクラスIDパラメータは最後に書き込まなければならない。よって、MFCコマンド・パラメータを書き込むためには、図20に概略を示す動作に従う。

20

#### 【0100】

図20に示すように、動作は、MFCローカル記憶アドレス・パラメータをMFCローカル記憶アドレス・チャンネルに書き込むこと（ステップ810）を含む。有効アドレスHighパラメータは、MFC有効アドレスHighチャンネルへ書き込まれる（ステップ820）。MFC有効アドレスLowまたはリスト・アドレス・パラメータは、MFC有効アドレスLowまたはリスト・アドレス・チャンネルへ書き込まれる（ステップ830）。MFC転送またはリスト・サイズ・パラメータは、MFC転送サイズまたはリスト・サイズ・チャンネルへ書き込まれる（ステップ840）。MFCコマンド・タグ・パラメータは、MFCコマンド・タグ識別子チャンネルへ書き込まれる（ステップ850）。上述のすべてのパラメータがそれぞれのチャンネルに書き込まれた後、MFCコマンド操作符号およびクラスIDパラメータが、MFC操作符号およびMFCクラスIDチャンネルへ書き込まれて（ステップ860）、動作は終了する。ステップ810～850は任意の順序で行われてもよく、ステップ860は他のパラメータのそれぞれのチャンネルへの書き込みの後に続くことが理解されるべきである。

30

#### 【0101】

MFCコマンド・パラメータは、MFCコマンド操作符号およびクラスIDパラメータの書き込みがMFCによって行われるまで、MFCコマンド・パラメータ・チャンネル内に保持される。MFCコマンド操作符号チャンネルおよびMFCクラスIDチャンネルへのチャンネル書き込み（wrc h）命令によって、MFCコマンド・パラメータ・チャンネル内に保持されたパラメータは、MFCコマンド・キューへ送られる。MFCコマンド・パラメータは、MFCコマンド自身がMFCコマンド・キューへ発行される前であれば任意の順序で書き込むことができる。MFCコマンド・パラメータ・チャンネルへ書き込まれた最後のパラメータの値が、エンキュー動作に使用される。

40

#### 【0102】

MFCコマンドがキューイングされた後、MFCパラメータの値は無効となり、次のMFCコマンドキューイング要求について再指定されなければならない。必要なMFCパラメータのすべて（すなわち、オプションのEAH以外のすべてのパラメータ）を指定しないと、MFCコマンド・キューの動作が不適切になりうる。

#### 【0103】

50

MFCコマンド操作符号チャンネルおよびMFCクラスIDチャンネルは、ハードウェアによってサポートされたMFCキューの数にハードウェアによって構成された最大カウントを有する。ソフトウェアは、電源投入後であってMFCプロキシ・コマンド・キューのパーズ後に、MFCコマンド操作符号チャンネルのチャンネル・カウントを、実施によってサポートされるMFCプロキシ・コマンド・キュー・スロットの数に初期化しなければならない。また、MFCコマンド操作符号チャンネルのチャンネル・カウントは、SPEプリエンティブ・コンテキスト・スイッチ上で退避および復元されなければならない。

#### 【0104】

##### MFCタグ・グループ・ステータス・チャンネル

10

上述のように、各コマンドは、5ビット識別子などのMFCコマンド・タグと称される識別子でタグ付けされる。同一の識別子を複数のMFCコマンドのために使用することができる。同一の識別子を有するコマンドの組は、タグ・グループと定義される。ソフトウェアは、タグ・グループ毎のキューイングされたすべてのコマンドの完了をチェックまたは待つために、使用することができる。加えて、MFCコマンド・タグは、MFC・DMAリスト・コマンドのチェックまたはそれを待つこと、ストール／通知フラグセットを有する要素への到達、MFC・DMAリスト・コマンドを再開するためのリスト要素の確認のために、ソフトウェアによって使用される。

#### 【0105】

MFCタグ・グループ・ステータス・チャンネルをまず説明し、その後、タグ・グループのステータスの判断と、MFC・DMAリスト・コマンドの完了の判断のための手順を説明する。

20

#### 【0106】

MFCタグ・グループ・ステータス・チャンネルは、MFC書き込みタグ・グループ・クエリ・マスク・チャンネルと、MFC読み出しタグ・グループ・クエリ・マスク・チャンネルと、MFC書き込みタグ・ステータス更新要求チャンネルと、MFC読み出しタグ・グループ・ステータス・チャンネルと、MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルと、MFC書き込みリスト・ストール／通知タグ確認チャンネルと、MFC読み出しアトミック・コマンド・ステータス・チャンネルとを含む。これらの各チャンネルを、以下に詳細に説明する。

30

#### 【0107】

##### MFC書き込みタグ・グループ・クエリ・マスク・チャンネル

MFC書き込みタグ・グループ・クエリ・マスク・チャンネルは、クエリまたは待ち動作に含まれるべきタグ・グループを選択するために使用される。MFC書き込みタグ・グループ・クエリ・マスク・チャンネルの詳細を図21に示す。

#### 【0108】

このチャンネルによって提供されるデータは、このチャンネルへ発行される後続のチャンネル書き込み(wrch)命令によって変更されるまで、MFCに保持される。したがって、データは、ステータス・クエリまたは待ち毎に再指定される必要はない。MFCタグ・ステータス更新要求が保留中の場合にマスクがソフトウェアによって修正されると、結果の意味合いは曖昧である。保留MFCタグ・ステータス更新要求は、このマスクの修正の前に常にキャンセルされなければならない。MFCタグ・ステータス更新要求は、「0」の値(すなわち、直接更新)をMFC書き込みタグ・ステータス更新要求チャンネルに書き込むことによってキャンセルすることができる。このチャンネルの現在の内容は、MFC読み出しタグ・グループ・クエリ・マスク・チャンネルを読み出す(rdch)ことによってアクセスすることができる。このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し(rchcnt)命令がこのチャンネルに送られると、カウントは常に「1」として返される。

40

#### 【0109】

##### MFC読み出しタグ・グループ・クエリ・マスク・チャンネル

50

MFC読み出しタグ・グループ・クエリ・マスク・チャンネルは、プロキシ・タグ・グループ・クエリ・マスク・レジスタの現在の値を読み出すために使用される。MFC読み出しタグ・グループ・クエリ・マスク・チャンネルの詳細を図22に示す。このチャンネルを読み出すと、MFC書き込みタグ・グループ・クエリ・マスク・チャンネルへ書き込まれた最後のデータを返す。プロキシ・タグ・グループ・クエリ・マスクのソフトウェア・シャドー・コピーを回避し、SPEコンテキスト退避/復元コマンド動作のために、このチャンネルを使用することができる。このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し(rchcnt)命令がこのチャンネルに送られると、カウントは常に「1」として返される。

【0110】

#### MFC書き込みタグ・ステータス更新要求チャンネル

MFC書き込みタグ・ステータス更新要求チャンネルは、MFCタグ・グループ・ステータスがMFC読み出しタグ・グループ・ステータス・チャンネルにおいて更新される時を制御する。MFC書き込みタグ・ステータス更新要求チャンネルの詳細を図23に示す。

【0111】

MFC書き込みタグ・ステータス更新要求チャンネルは、ステータスを即時に更新、または条件の発生で更新を指定することができる。条件の発生とは、例えば、イネーブルなMFCタグ・グループ完了が「未処理の処理なし」ステータスを有する場合に更新される、またはすべてのイネーブルなMFCタグ・グループ「未処理の処理なし」ステータスを有するときのみ更新されるといったことである。このチャンネルへのチャンネル書き込み(wrch)命令は、MFC読み出しタグ・グループ・ステータス・チャンネルからのチャンネル読み出し(rdch)が発生するまでに、発生しなければならない。

【0112】

MFC書き込みタグ・ステータス更新要求は、タグ・グループ・マスクの設定後、および当該タグ・グループに対するコマンドの発行後に行われなければならない。タグ・グループに対するコマンドがMFC書き込みタグ・ステータス更新要求を発行する前に完了させて、更新ステータス条件を満たすと、ステータスは待たずに戻る。MFC書き込みタグ・ステータス更新要求チャンネルに書き込むことによって最初にステータス更新を要求することをせずにMFC読み出しタグ・グループ・ステータス・チャンネルから読み出すと、ソフトウェアによって引き起こされるデッドロックという結果となる。

【0113】

以前のMFCタグ・ステータス更新要求のキャンセルは、直接更新ステータス要求をMFC書き込みタグ・ステータス更新要求チャンネルへ発行し、MFC書き込みタグ・ステータス更新要求チャンネルに関するカウントを「1」の値が返るまで読み出し、MFC読み出しタグ・グループ・ステータス・チャンネルから読み出して、所望の不要な結果を破棄および判断することによって行うことができる。

【0114】

介在ステータス読み出し要求なしの条件付きの更新要求は、予測できないタグ・ステータスを返すこととなる。予測できない結果を回避するために、ソフトウェア対は、直接更新要求を介して要求キャンセルが行われていなければ、タグ・ステータスへの読み出しを伴うタグ・ステータス更新を要求する。

【0115】

特権的ソフトウェアは、このチャンネルについてのカウントを「1」に初期化する。チャンネル書き込み(wrch)命令がこのチャンネルに対して発行されると、このチャンネルについてのカウントは「0」に設定される。MFCがタグ・ステータス更新要求を受信すると、カウントは「1」に設定される。このチャンネルは、最大カウント「1」で書き込みブロッキングイネーブルされる。

【0116】

#### MFC読み出しタグ・グループ・ステータス・チャンネル

10

20

30

40

50



MFC読み出しタグ・グループ・ステータス・チャンネルは、最後のタグ・グループ・ステータス更新要求からのタグ・グループのステータスを含む。MFC読み出しタグ・グループ・ステータス・チャンネルの詳細を図24に示す。

【0117】

タグ・グループ・ステータス更新の時にイネーブルなタグ・グループのステータスのみが有効である。タグ・グループ・ステータス更新の時にディスエーブルなタグ・グループに対応するビット位置は、「0」に設定される。

【0118】

MFC書き込みタグ・ステータス更新要求チャンネルは、このチャンネルからの読み出し前に要求されなければならない。それを行わないと、ソフトウェアによって引き起こされるデッドロックという結果となる。

【0119】

MFC読み出しタグ・グループ・ステータス・チャンネルへ送られたチャンネル・カウント読み出し(rchcnt)命令は、ステータスがまだ使用可能でなければ「0」を、ステータスが使用可能であれば「1」を返す。この命令は、MFC読み出しタグ・グループ・ステータス・チャンネルが読み出される場合にSPUがストールするのを回避するために使用することができる。ソフトウェアは、このチャンネルについてのカウントを「0」の値に初期化する。このチャンネルは、最大カウント「1」で読み出しブロッキングイネーブルされる。

【0120】

MFC読み出しリスト・ストール/通知タグ・ステータス・チャンネル

MFC読み出しリスト・ストール/通知タグ・ステータス・チャンネルの詳細を図25に示す。上述のように、MFCリスト・コマンドについてのリスト要素は、ストール/通知タグを含む。フラグがリスト要素に設定されている場合には、MFCは、MFCリスト・コマンド、すなわち、DMAリスト・コマンドを実行するのを停止して、このチャンネルのMFCリスト・コマンドのタグ・グループに対応するビットを設定する。また、このチャンネルに関連したカウントが「1」に設定される。MFCリスト・コマンドは、MFC書き込みリスト・ストール/通知タグ確認チャンネルへタグ値を書き込むことによって確認されるまでタグ値を書き込むことによって確認されるまで、ストールされたままである。

【0121】

MFCリスト・ストール/通知ファシリティは、DMAリストの実行がある特定の点に到達した場合にプログラムが通知を受けたい場合に有用である。これは、ストールされたリスト要素に続くリスト要素(転送サイズまたは有効アドレス)をアプリケーションが動的に変更したい場合にも有用である。また、リスト要素は、その転送サイズを「0」に設定することによってスキップすることができる。ハードウェアは、ストール/通知要素を越えてリスト要素を先取りすることができない。

【0122】

特権的ソフトウェアは、MFC読み出しリスト・ストール/通知タグ・ステータス・チャンネルのカウントを0に初期化する。ソフトウェアは、どのタググループがこのチャンネルの最後の読み出し以降ストールしているかコマンドを有するかを、このチャンネルの内容を再び読み出すことによって判断することができる。チャンネル読み出し(rdch)命令をこのチャンネルに発行すると、すべてのビットを0にリセットし、このチャンネルに対応するカウントを「0」に設定する。したがって、「1」に設定されたストール/通知フラグを含む未処理のリストと、ストールされたコマンドとがないチャンネル読み出し(rdch)命令を発行すると、ソフトウェアによって引き起こされるデッドロックという結果となる。

【0123】

ストールされているタグ・グループがない場合にこのチャンネル上にチャンネル読み出し(rdch)命令を発行すると、ストール/通知フラグが設定されたリスト要素が生じ

10

20

30

40

50

るまで、SPU実行がストールする結果となる。また、ソフトウェアは、このチャンネルに関連したカウントを読み出して (rchcnt)、SPUイベント・ファシリティと共に、ストール／通知フラグが設定されたMFCリスト要素が生じることを判断するために使用することができる。MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルへ送られたチャンネル・カウント読み出し (rchcnt) 命令は、このチャンネルの最後の読み出し以降新たにストールされたMFCリスト・コマンドがない場合は、「0」を返す。このチャンネルは、読み出しブロッキングであり、最大カウント「1」を有する。

#### 【0124】

##### MFC書き込みリスト・ストール／通知タグ確認チャンネル

MFC書き込みリスト・ストール／通知タグ確認チャンネルの詳細を図26に示す。MFC書き込みリスト・ストール／通知タグ確認チャンネルは、ストール／通知フラグが設定されたリスト要素でストールされているMFCリスト・コマンドを含むタグ・グループを受け取り確認するために使用される。MFCタグ・グループをこのチャンネルに書き込むことによって、タグ・グループは確認される。書き込みの後、このチャンネルに書き込まれた値と一致するタグ・グループのすべてのストールされたMFCリスト・コマンドが再開される。

#### 【0125】

MFCリスト・ストール／通知ファシリティは、DMAリストの実行がある特定の点に到達した場合にプログラムが通知を受けたい場合に有用である。これは、ストールされたリスト要素に続くリスト要素 (転送サイズまたは有効アドレス) をアプリケーションが動的に変更したい場合にも有用である。また、リスト要素は、その転送サイズを「0」に設定することによってスキップすることができる。ハードウェアは、ストール／通知要素を越えてリスト要素を先取りすることができない。

#### 【0126】

ストール／通知状態によって現在ストールされていないタグ・グループを確認することは、定義されていない。そうすると、MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネル内の無効なステータスという結果となる。整合性のために、この状態は、ノー・オペレーション (no-op) として扱われる。

#### 【0127】

このチャンネルは、非ブロッキング・チャンネルであり、関連カウントを有しない。チャンネル・カウント読み出し (rchcnt) 命令をこのチャンネルへ送る度に、カウントは常に「1」として返される。

#### 【0128】

##### MFC読み出しアトミック・コマンド・ステータス・チャンネル

MFC読み出しアトミック・コマンド・ステータス・チャンネルの詳細を図27に示す。MFC読み出しアトミック・コマンド・ステータス・チャンネルは、最後に完了した直接MFCアトミック更新コマンドのステータスを含む。アトミック・コマンドとは、MFC・SPUキュー内の他のコマンドを待たずに実行され、かつ、MFC・SPUキュー内の他のコマンドとは独立して実行されるコマンドである。MFCによってサポートされているアトミック・コマンドは、getllar (get lock line reserve)、putllc (put lock line conditional)、putlluc (put lock line unconditional)、putqluc (put queued lock line unconditional) の4つである。これらのコマンドは、「ロック」を解除するためにソフトウェアによって従来使用されたキャッシュ可能な記憶命令と同様の機能を行う。putllucおよびputqlucコマンドの違いは、putqlucコマンドは、MFC・SPUコマンド・キュー内の他のMFCコマンドの後ろにタグ付けまたはキューイングされるが、putllucコマンドは、即座に実行される点である。

#### 【0129】

10

20

30

40

50

putqlluc コマンドはタグ付けされて、黙示のタグ別フェンスを有するので、MFC・SPU コマンド・キュー内に既にある同じタグ・グループ内のすべての他のコマンドに対して命令される。gellar, putllc, および putlluc コマンドはタグ付けされていないので、これらは即座に実行される。gellar, putllc, および putlluc コマンドは即座に実行されるが、それでも、これらのコマンドは、MFC・SPU コマンド・キュー内の使用可能なスロットを必要とする。MFC・SPU コマンド・キュー内での他のコマンドとの順序付けは想定されてはならない。各 gellar, putllc, および putlluc コマンドの発行後、ソフトウェアは、MFC 読み出しアトミック・コマンド・ステータス・チャンネルからの読み出しを発行して、コマンドの完了を検証しなければならない。直接アトミック・コマンドを発行する前にチャンネル読み出し (rdch) 命令をこのチャンネルに発行すると、ソフトウェアによって引き起こされるデッドロックという結果となる。

10

#### 【0130】

ソフトウェアは、このチャンネルに関連したチャンネル・カウント (rchcnt) を読み出して、即座のアトミック MFC コマンドが完了したかどうかを判断することができる。「0」の値が返る場合には、即座のアトミック MFC コマンドは完了していない。「1」の値が返る場合には、即座のアトミック MFC コマンドは完了しており、このチャンネルを読み出す (rdch) ことによって、ステータスを使用できる。

#### 【0131】

MFC 読み出しアトミック・コマンド・ステータス・チャンネルからの読み出し (rdch) の後には、即座のアトミック MFC コマンドが常に続かなくてはならない。MFC 読み出しアトミック・コマンド・ステータス・チャンネルからの介在読み出しをすることなく複数のアトミック MFC コマンドを行うと、誤ったステータスという結果となる。

20

#### 【0132】

特権的ソフトウェアは、このチャンネルのカウントを「0」に初期化する。このチャンネルは、最大カウント「1」の読み出しブロッキングである。このチャンネルの内容は、読み出されるとクリアされる。後続の即座の MTC アトミック更新コマンドは、先行する MFC コマンドの状態を上書きする。

#### 【0133】

上記の MFC タグ・グループ・ステータス・チャンネルは、タグ・グループのステータスの判断を促進すると共に、MFC・DMA リスト・コマンド完了を判断するために使用される。タグ・グループのステータスを判断するための以下の3つの基本的な手順がサポートされている。すなわち、MFC 読み出しタグ・グループ・ステータス・チャンネルをポーリングすることと、タグ・グループ更新を待つことまたはイベントを待つことと、タグ・グループ・ステータス更新イベントに割り込むことである。MFC コマンドの完了および MFC コマンドのグループの完了についてのポーリングのための基本的な手順は、図 28 に示すとおりである。

30

#### 【0134】

図 28 に示すように、いずれの保留タグ・ステータス更新要求もクリアされる (ステップ 1010)。これは、MFC 書き込みタグ・ステータス更新要求チャンネルに 0 を書き込み、MFC 書き込みタグ・ステータス更新要求チャンネルに関連したチャンネル・カウントを「1」の値が返るまで読み出し、MFC 読み出しタグ・グループ・ステータス・チャンネルを読み出してタグ・ステータス・データを廃棄することによって達成されてもよい。

40

#### 【0135】

当該タグ・グループは、その後、MFC 書き込みタグ・グループ・クエリ・マスク・チャンネルに適切なマスク・データを書き込む (ステップ 1020)。その後、例えば、MFC 書き込みタグ・ステータス更新要求チャンネルに「0」の値を書き込むことによって、直接タグステータス更新が要求される (ステップ 1030)。

#### 【0136】

50

その後、MFC読み出しタグ・グループ・ステータス・チャンネルの読み出しが行われる（ステップ1040）。返されるデータは、タグ・グループ・マスクを適用した状態の各タグ・グループの現在のステータスである。当該タグ・グループの追加があるかどうかの判断が行われる（ステップ1050）。追加があれば、動作はステップ1030に戻る。相でなければ、動作は終了する。

#### 【0137】

タグ・グループ更新またはイベント（1つ以上のタグ・グループ完了）を待つための基本的な手順を図29に示す。図示のように、動作は、いずれの保留タグ・ステータス更新要求もクリアすることによって開始する（ステップ1110）。これは、例えば、MFC書き込みタグ・ステータス更新要求チャンネルに「0」を書き込み、MFC書き込みタグ・ステータス更新要求チャンネルに関連したチャンネル・カウントを「1」の値が返るまで読み出し、MFC読み出しタグ・グループ・ステータス・チャンネルを読み出してタグ・ステータス・データを廃棄することによって達成されてもよい。

#### 【0138】

条件付タグステータス更新は、MFC書き込みタグ・ステータス更新要求チャンネルに「01」または「10」の値を書き込むことによって要求される（ステップ1120）。「01」の値は、タグ・グループ更新内の任意のタグ・グループの完了を指定する。「10」の値は、すべてのイネーブルなタグ・グループはSPUタグ・グループステータス更新という結果になるように干渉しなければならないということを指定する。

#### 【0139】

その後、MFC読み出しタグ・グループ・ステータス・チャンネルを読み出して、ステップ1120において指定された特定のタグ・イベントを待つ（ステップ1130）。この読み出しにより、ステップ1120において指定された条件が満たされるまでSPUの実行がストールする。代わりに、特定のタグ・イベントのためのポーリングまたは待ちのためのMFC読み出しタグ・グループ・ステータス・チャンネルに関連したカウントの読み出しが行われてもよい（ステップ1132）。

#### 【0140】

返ってきたカウントが「1」の値を有するかどうかの判断が行われる（ステップ1140）。もしそうでなければ、動作はステップ1132へ戻る。カウントが「1」である場合には、MFC読み出しタグ・グループ・ステータス・チャンネルを読み出して、1つのタグ・グループまたは複数のタグ・グループが完了するかどうかを判断する（ステップ1150）。その後、動作は終了する。

#### 【0141】

条件付タグ・イベントを待つこと、またはそれをポーリングすることの代わりとしては、SPUイベント・ファシリティを使用することが挙げられる。この手順は、典型的には、アプリケーションが複数のイベントのうちの1つが生じるのを待っているか、コマンド完了を待っている間に他のワークを行うことができる場合に使用される。この手順は、図30に示すとおりである。

#### 【0142】

図30に示すように、任意の保留タグ・ステータス更新要求がクリアされる（ステップ1210）。上述のように、これは、例えば、MFC書き込みタグ・ステータス更新要求チャンネルに「0」を書き込み、MFC書き込みタグ・ステータス更新要求チャンネルに関連したチャンネル・カウントを「1」の値が返るまで読み出し、MFC読み出しタグ・グループ・ステータス・チャンネルを読み出してタグ・ステータス・データを廃棄することによって達成されてもよい。

#### 【0143】

1つのタグ・グループまたは複数のタグ・グループを選択する（ステップ1220）。SPU書き込みイベント確認チャンネルに「1」の値を書き込む（w r c h）ことによって、いずれの保留タグ・ステータス更新イベントもクリアする（ステップ1230）。MFCタグ・グループ・ステータス更新イベントを、SPU書き込みイベント・マスク・チ

チャンネルに「1」を書き込むことによって、アンマスクする（ステップ1240）。その後、SPU読み出しイベント・ステータス・チャンネルを読み出して、イネーブルなイベントが生じるのを待つ（ステップ1250）。この読み出しにより、イネーブルなイベントが生じるまで、SPUの実行はストールする。代わりに、SPU読み出しイベント・ステータス・チャンネルを読み出して、カウントが「1」として返るまで、特定のタグ・イベントをポーリングまたは待ってもよい。

#### 【0144】

SPU読み出しイベント・ステータス・チャンネルを読み出して、イネーブルなイベントが生じたかどうかについての判断が行われる（ステップ1260）。生じていなかった場合、動作はステップ1250へ戻る。イネーブルなイベントが生じた場合は、MFC読み出しタグ・グループ・ステータス・チャンネルを読み出して、どのタグまたはタグ・グループがイベントを生じさせたかを判断する（ステップ1270）。その後、動作は終了する。

10

#### 【0145】

MFC・DMAリスト・コマンドがストール／通知フラグが設定されたリスト要素に到達したかを判断するための3つの基本的な手順がサポートされている。すなわち、MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルをポーリングすることと、MFC・DMAリスト・コマンド・ストール／通知イベントを待つことと、MFC・DMAリスト・コマンド・ストール／通知イベントに割り込むことである。MFC・DMAリスト・コマンドがストール／通知フラグが設定されたリスト要素に到達したかを判断するためのポーリングのための基本的な手順は、図31に示すとおりである。

20

#### 【0146】

図31に示すように、動作は、ストール／通知フラグが設定されたリスト要素を有するMFC・DMAリスト・コマンドの発行によって開始する（ステップ1310）。MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルに関連したカウントを、「1」の値が返るまで読み出す（rchcnt）（ステップ1320）。「1」という値が返る場合（ステップ1330）、MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルを読み出す（rdch）（ステップ1340）。返ったデータは、このチャンネルの最終読み出し以降ストール／通知フラグが設定されたリスト要素に到達した各タググループの現在のステータスである。

30

#### 【0147】

1つのタグ・グループまたは複数の当該タグ・グループがストール／通知フラグが設定されたリスト要素に到達したかどうかについての判断がなされる（ステップ1350）。到達していなければ、タググループまたは複数の当該タグ・グループがストール／通知フラグが設定されたリスト要素に到達するまで、動作はステップ1340へ戻る。

#### 【0148】

1つのタグ・グループまたは複数の当該タグ・グループがストール／通知フラグが設定されたリスト要素に到達した場合には、MFC書き込みリスト・ストール／通知タグ確認チャンネルに、ストールしたタグ・グループに対応するタグ・グループ番号を書き込んで（wrch）、MFC・DMAリスト・コマンドを再開する（ステップ1360）。その後、動作は停止する。

40

#### 【0149】

ストール／通知フラグが設定されたリスト要素に到達するためにMFC・DMAリスト・コマンドを待つ基本的な手順は、図32に示すとおりである。図示のように、動作は、ストール／通知フラグが設定されたリスト要素を有するMFC・DMAリスト・コマンドが発行されることで開始する（ステップ1410）。MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルの読み出し（rdch）が行われる（ステップ1420）。返されたデータは、このチャンネルの最終読み出し以降のストール／通知フラグが設定されたリスト要素に到達した各タグ・グループの現在のステータスである。この読み出しにより、MFC・DMAリスト・コマンドがストール／通知フラグが設定されたリス

50

ト要素に到達するまで、SPUはストールする。

#### 【0150】

1つのタグ・グループまたは複数の当該タグ・グループがストール／通知フラグが設定されたリスト要素に到達したかどうかについての判断がなされる（ステップ1430）。対応ビットが、返りデータに設定される。読み出し毎にビットがリセットされるので、ソフトウェアは、複数のタグ・グループがストールするのを待つ間に、タグ・グループの蓄積を行う。

#### 【0151】

到達していなければ、単数のタグ・グループまたは複数の当該タグ・グループがストール／通知フラグが設定されたリスト要素に到達するまで、動作はステップ1420に戻る。そうでなければ、MFC書き込みリスト・ストール／通知タグ確認チャンネルに対して、ストールされたタグ・グループに対応するタグ・グループ番号の書き込み（wrc h）が行われて、MFC・DMAリスト・コマンドを再開させる（ステップ1440）。

#### 【0152】

ストール／通知タグ・グループ・ステータスを待つこと、またはそれをポーリングすることの代わりとしては、SPUイベント・ファシリティを使用することが挙げられる。この手順は、典型的には、MFC・DMAリスト・コマンドが実行している間に、他のワークがSPUプログラムによって事項できる場合に使用される。この手順は、例えば、図33にその概略を示すとおりである。

#### 【0153】

図33に示すように、手順は、どの保留MFC・DMAリスト・コマンド・ストール／通知イベントもクリアすることによって開始する（ステップ1510）。これは、例えば、SPU書き込みイベント確認チャンネルに「1」の値を書き込む（wrc h）ことによって達成されてもよい。MFC・DMAリスト・コマンド・ストール／通知イベントは、「1」をSPU書き込みイベント・マスク・チャンネルのSnビットに書き込むことによってイネーブルされる（1520）。ストール／通知フラグが設定されたリスト要素を有するMFC・DMAリスト・コマンドが発行される（ステップ1530）。

#### 【0154】

イネーブルなイベントが生じるのを待つために、SPU読み出しイベント・ステータス・チャンネルからの読み出し（rdch）が行われてもよい（1540）。この読み出しにより、イネーブルなイベントが生じるまでSPUの実行がストールされる。代わりに、カウントが「1」に戻るまで特定のタグ・イベントをポーリングするために、SPU読み出しイベント・ステータス・チャンネルに関連したカウントの読み出し（rchcnt）が行われてもよい。

#### 【0155】

イネーブルなイベントが生じたかどうかについての判断が行われる（ステップ1550）。生じていない場合には、動作はステップ1540に戻る。イネーブルなイベントが生じた場合には、DMAリスト・ストール／通知イベントが生じているかどうかについての判断が行われる（ステップ1560）。DMAリスト・ストール／通知イベントが生じていない場合には、動作はステップ1540に戻る。

#### 【0156】

DMAリスト・ストール／通知イベントが生じている場合には、MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルからの読み出しが行われて、どの1つのタグ・グループまたは複数のグループがイベントを生じさせたかを判断する（ステップ1570）。MFC書き込みリスト・ストール／通知タグ確認チャンネルに対して、ストールされたタグ・グループに対応するタグ・グループ番号の書き込み（wrc h）が行われて、MFC・DMAリスト・コマンドを再開させる（ステップ1580）。

#### 【0157】

##### MFC書き込みマルチソース同期要求チャンネル

MFC書き込みマルチソース同期要求チャンネルの詳細を図34に示す。MFC書き込

10

20

30

40

50

みマルチソース同期要求チャンネルは、MFCマルチソース同期ファシリティの一部であり、MFCに対して、関連MFCへ送られた未処理の転送を追跡することを開始させる。MFCマルチソース同期ファシリティは、プロセッサまたは装置がメイン記憶アドレスドメインから同期を制御できるようにするMFCマルチソース同期レジスタと、SPUがローカル記憶アドレスドメインから同期を制御できるようにするMFC書き込みマルチソース同期要求チャンネル(MFC\_WrMSSyncReq)とを含む。

#### 【0158】

同期は、MFC書き込みマルチソース同期要求チャンネルへの書き込みによって要求されてもよい。要求された同期が完了すると、チャンネル・カウンタは「1」に設定し戻され、このチャンネルに書き込まれたデータは無視される。このチャンネルに対する第2の書き込みにより、SPUは、第1の書き込みによって追跡された未処理の転送が完了するまでストールする結果となる。

10

#### 【0159】

MFC書き込みマルチソース同期要求チャンネルを使用するには、プログラムが、MFC書き込みマルチソース同期要求チャンネルに対して書き込みを行い、その後、MFC書き込みマルチソース同期要求チャンネルが使用可能になるのを待つ。すなわち、チャンネル・カウンタが「1」に設定し戻されるのを待つ。ソフトウェアは、このチャンネルについてのカウンタを「1」の値に初期化する。このチャンネルは、最大カウンタ「1」で書き込みブロッキング・イネーブルされる。

#### 【0160】

20

##### メールボックス・ファシリティ

MFCとの通信のためのチャンネルに加えて、本発明のチャンネル・インターフェースは、SPUにおいて提供されるメールボックス・ファシリティとの通信のためのチャンネルをさらに提供する。MFCは、SPUと他のプロセッサおよび装置との間のメールボックス・キューのセットを提供する。各メールボックス・キューは、割り当てられたSPUチャンネルと、対応のMMIOレジスタとを有する。SPUソフトウェアは、SPUチャンネル命令を使用してメールボックス・キューをアクセスする。他のプロセッサおよび装置は、MMIOレジスタのうちの1つを使用してメールボックス・キューをアクセスする。キューに加えて、MFCは、キュー・ステータスと、メールボックス割り込みと、メールSPUイベント通知とをメールボックス用に提供する。MMIOレジスタ、チャンネル、ステータス、割り込み、メールボックス・キュー、およびイベントを総称して、「メールボックス・ファシリティ」と称する。上述のように、メールボックス・ファシリティは、図2のMFCレジスタ・ユニット240内に提供される。

30

#### 【0161】

SPUから他のプロセッサまたは他の装置へ情報を送るために、MFCによって2つのメールボックス・キュー、すなわち、SPUアウトバウンド・メールボックス・キューと、SPUアウトバウンド割り込みメールボックス・キューとが提供される。これらのメールボックス・キューは、短いメッセージをPPEへ送るためのものである(例えば、復帰コードまたはステータス)。チャンネル書き込み(wrch)命令を使用してSPUによってこれらのキューのうちの1つに書き込まれたデータは、対応するMMIOレジスタを読み出すことによって、どのプロセッサまたは装置にも使用可能である。

40

#### 【0162】

SPU書き込みアウトバウンド・割り込みメールボックス・チャンネルへ送られた書き込み(wrch)命令によっても、システム内のプロセッサまたは他の装置へ割り込みを送ることができる。これらのキューのいずれか(SPUアウトバウンド・メールボックス・キューまたはSPUアウトバウンド割り込みメールボックス・キュー)からのMMIO読み出しによって、SPUイベントを設定することができ、それによってSPU割り込みが生じる。

#### 【0163】

SPUへ情報を送るために、外部プロセッサまたは他の装置のいずれかに対して、1つ

50

のメールボックス・キュー、すなわち、SPUインバウンド・メールボックス・キューが提供される。このメールボックス・キューは、PPEによって書き込まれるものである。しかしながら、他のプロセッサ、SPU、または他の装置がこのメールボックス・キューを使用してもよい。MMIO書き込みを使用してプロセッサまたは他の装置によってこのキューに書き込まれるデータは、SPU読み出しインバウンド・メールボックス・チャンネルを読み出すことによって、SPUが使用可能である。SPUインバウンド・メールボックス・レジスタへのMMIO書き込みによって、SPUイベントを設定することができ、それによってSPU割り込みが生じる。

#### 【0164】

SPUアウトバウンド割り込みメールボックス・レジスタは、対応のSPUアウトバウンド・メールボックス・キューから32ビットのデータを読み出すために使用される。SPUアウトバウンド割り込みメールボックス・レジスタは、SPUアウトバウンド・メールボックス・キューヘデータを書き込むための対応SPU書き込みアウトバウンド・メールボックス・チャンネルを有する。SPUアウトバウンド・メールボックス・キューへ送られたチャンネル書き込み(wrch)命令は、当該命令内で指定された32ビットのデータを、他のプロセッサまたは他の装置が読み出すことができるように、SPUアウトバウンド・メールボックス・キューへロードする。SPUアウトバウンド・メールボックス・キューが満杯の場合には、SPUは、このメールボックス・レジスタからのMMIO読み出しが生じるまで、このキューへ送られるチャンネル書き込み(wrch)命令でストールする。

#### 【0165】

このレジスタのMMIO読み出しは、SPUによって書き込まれた順序で情報を常に返す。空のSPUアウトバウンド・メールボックス・キューからの読み出しで返された情報は、定義されていない。SPUアウトバウンド・メールボックス・キュー内のエントリの数(またはキュー深度)は、実施に依存する。

#### 【0166】

SPUメールボックス・ステータス・レジスタのMMIO読み出しは、メールボックス・キューのステータスを返す。SPUアウトバウンド・メールボックス・キュー内の有効なキュー・エントリ数は、SPUメールボックス・ステータス・レジスタのSPU\_Out\_Mbox\_Countフィールド内に与えられる。SPUメールボックス・ステータス・レジスタのMMIO読み出しは、保留SPUアウトバウンド・メールボックス使用可能イベントを設定する。メールボックス・キューに残っているデータ量が実施に依存する閾値を下回り、この条件がイネーブル(すなわち、SPU\_WrEventMask[Le]が「1」に設定される)の場合、SPU読み出しイベント・ステータス・チャンネルが更新され(すなわち、SPU\_RdEventStat[Le]が「1」に設定される)、そのチャンネル・カウントが「1」に設定される。これにより、SPUアウトバウンド割り込みメールボックス使用可能イベントが生じる。

#### 【0167】

SPUインバウンド・メールボックス・レジスタは、対応のSPUインバウンド・メールボックス・キューへ32ビットのデータを書き込むために使用される。SPUインバウンド・メールボックス・キューは、キューからデータを読み出すための対応SPU読み出しインバウンド・メールボックス・チャンネルを有する。SPU読み出しインバウンド・メールボックス・チャンネルのチャンネル読み出し(rdch)命令は、チャンネル読み出し(rdch)命令によって指定されたSPUレジスタにSPUインバウンド・メールボックス・キューからの32ビットのデータをロードする。SPUは、空のメールボックスから読み出すことはできない。SPUインバウンド・メールボックス・キューが空の場合には、SPUは、メールボックスにデータが書き込まれるまで、このチャンネルに対するチャンネル読み出し(rdch)命令でストールする。このチャンネルに対するチャンネル読み出し(rdch)命令は、PPEまたは他のプロセッサまたは装置によって書き込まれた順序で情報を常に返す。

10

20

30

40

50



## 【0168】

キュー内のエントリの数（またはキュー深度）は、実施に依存する。SPUメールボックス・ステータス・レジスタのMMIO読み出しは、メールボックス・キューの状態を返す。SPUメールボックス・キュー内の使用可能なキュー位置の数は、SPUメールボックス・ステータス・レジスタのSPU\_In\_Mbox\_Countフィールド（すなわち、SPU\_Mbox\_Stat[SPU\_In\_Mbox\_Count]）内で与えられる。

## 【0169】

ソフトウェアは、SPUメールボックスをオーバランするのを回避するために、SPU\_In\_Mboxへ書き込む前にSPUメールボックス・ステータス・レジスタをチェックする。SPUインバウンド・メールボックス・レジスタのMMIO書き込みにより、保留SPUメールボックス・イベントが設定される。イネーブル（すなわち、SPU\_Wr\_EventMask[Mbox] = 「1」）の場合は、SPU読み出しイベント・ステータス・チャンネルは更新され、そのチャンネル・カウントは「1」に設定されて、それにより、SPUインバウンド・メールボックス使用可能イベントが生じる。

## 【0170】

SPUメールボックス・ステータス・レジスタは、SPUと、対応するSPE内のPPEとの間のメールボックス・キューの現在の状態を含む。このレジスタを読み出すことは、メールボックス・キューの状態に影響を与えない。

## 【0171】

SPUメールボックス・チャンネル

上述のように、MFCによって提供されるメールボックス・ファシリティは、SPU書き込みアウトバウンド・メールボックス・チャンネルと、SPU書き込みアウトバウンド割り込みメールボックス・チャンネルと、SPU読み出しインバウンド・メールボックス・チャンネルとを含む複数のSPUメールボックス・チャンネルを含む。これらのSPUメールボックス・チャンネルは、ブロッキング、すなわち、チャンネルが満杯の場合にSPUをストールする（書き込みブロッキング）、または、データが使用可能でない場合にSPUをストールする（読み出しブロッキング）として定義される。チャンネルのブロッキング方法は、実行する他のワークがアプリケーションにない場合に、電力節約のために非常に有利である。元来、プロセッサは、空間が開放されるまで、またはデータが使用可能となるまで、低電力状態に置かれてもよい。

## 【0172】

これらのチャンネルはブロッキングであるので節電上の利点を得る一方で、これらのチャンネルをアクセスすると、SPUが無期限にストールしてしまう場合がある。ソフトウェアは、後述のようにSPUイベント・ファシリティを使用して、またはメールボックス・チャンネルに関連したチャンネル・カウントを読み出すことによって、SPUのストールを回避することができる。

## 【0173】

SPU書き込みアウトバウンド・メールボックス・チャンネル

図35は、本発明の例示的な一実施形態に係るSPU書き込みアウトバウンド・メールボックス・チャンネルの詳細を示す。このチャンネルへ送られたチャンネル書き込み（wrch）命令は、SPU書き込みアウトバウンド・メールボックス・キューへデータを書き込む。このチャンネルにSPUによって書き込まれたデータは、SPUアウトバウンド・メールボックス・レジスタのMMIO読み出しのために使用可能である。また、このチャンネルへのチャンネル書き込み（wrch）命令により、関連チャンネル・カウントは「1」デクリメントされる。満杯のSPU書き込みアウトバウンド・メールボックス・キューへの書き込みは、SPU書き込みアウトバウンド・メールボックス・レジスタが読み出されてSPU書き込みアウトバウンド・メールボックス・キューが開放されるまで、SPUの実行をストールさせる。

## 【0174】

10

20

30

40

50

ストール状態を回避するために、このチャンネルに関連したチャンネル・カウントを読み出して、チャンネル書き込みを発行する前に、S P U書き込みアウトバウンド・メールボックス・キュー内にスロットがあることを保証することができる。代わりに、S P Uアウトバウンド・メールボックス使用可能イベントを使用して、S P U書き込みアウトバウンド・メールボックス・キュー内のスロットの使用可能性を、満杯と判断した場合に知らせることもできる。

#### 【0175】

S P U書き込みアウトバウンド・メールボックス・キューが満杯の場合は、このチャンネルに関連したチャンネル・カウントの読み出しは「0」の値を返す。非ゼロの値は、S P U書き込みアウトバウンド・メールボックス・キュー内の空いている32ビットワード数を示す。 10

#### 【0176】

特権的ソフトウェアは、S P U書き込みアウトバウンド・メールボックス・チャンネルのカウントをS P U書き込みアウトバウンド・メールボックス・キューの深度に初期化する。このチャンネルは、書き込みブロッキングである。このチャンネルについての最大カウントは、実施に依存しており、S P U書き込みアウトバウンド・メールボックス・キューの深度（すなわち、使用可能なスロット数）でなければならない。

#### 【0177】

##### S P U書き込みアウトバウンド割り込みメールボックス・チャンネル

図36は、本発明の例示的な一実施形態に係るS P U書き込みアウトバウンド割り込みメールボックス・チャンネルの詳細を示す。このチャンネルへのチャンネル書き込み（w r c h）命令は、S P U書き込みアウトバウンド割り込みメールボックス・キューヘデータを書き込む。このチャンネルにS P Uによって書き込まれたデータは、S P Uアウトバウンド割り込みメールボックス・レジスタのM M I O読み出しに使用可能となる。 20

#### 【0178】

また、このS P U書き込みアウトバウンド・メールボックス・チャンネルへのチャンネル書き込み（w r c h）命令により、関連チャンネル・カウントは「1」デクリメントされる。満杯のS P U書き込みアウトバウンド割り込みメールボックス・キューへの書き込みは、S P Uアウトバウンド割り込みメールボックス・レジスタが読み出されて、S P U書き込みアウトバウンド割り込みメールボックス・キューを開放するまで、S P Uの実行をストールさせる。 30

#### 【0179】

ストール状態を回避するために、このチャンネルに関連したチャンネル・カウントを読み出して、チャンネル書き込みを発行する前に、S P U書き込みアウトバウンド割り込みメールボックス・キュー内にスロットがあることを保証することができる。代わりに、S P Uアウトバウンド割り込みメールボックス使用可能イベントを使用して、S P U書き込みアウトバウンド割り込みメールボックス・キュー内のスロットの使用可能性を、満杯と判断した場合に知らせることもできる。また、S P U書き込みアウトバウンド・メールボックス・チャンネルへのチャンネル書き込み（w r c h）命令により、プロセッサまたは他の装置へ割り込みが送られる。割り込みおよび左記に発行されたM F Cコマンドの順序はない。 40

#### 【0180】

S P U書き込みアウトバウンド割り込みメールボックス・キューが満杯の場合に、このチャンネルに関連したチャンネル・カウントの読み出しは、「0」の値を返す。非ゼロのカウント値は、このキュー内の空いている32ビットワード数を示す。

#### 【0181】

特権的ソフトウェアは、このチャンネルのカウントをS P U書き込みアウトバウンド割り込みメールボックス・キューの深度に初期化する。このチャンネルは、書き込みブロッキングである。このチャンネルについての最大カウントは、実施に依存しており、S P U書き込みアウトバウンド割り込みメールボックス・キューの深度（すなわち、使用可能な 50

スロット数) でなければならない。

#### 【0182】

##### S P U読み出しインバウンド・メールボックス・チャンネル

図37は、本発明の例示的な一実施形態に係るS P U読み出しインバウンド・メールボックス・チャンネルの詳細を示す。このチャンネルからの読み出しは、S P U読み出しインバウンド・メールボックス・キュー内の次のデータを返す。プロセッサまたは装置がS P U読み出しインバウンド・メールボックス・レジスタへの書き込みを発行することによって、S P U読み出しインバウンド・メールボックス・キューにデータが置かれる。

#### 【0183】

S P U読み出しインバウンド・メールボックス・チャンネルからの読み出しにより、関連チャンネル・カウントは「1」デクリメントされる。空のメールボックスの読み出しは、S P U読み出しインバウンド・メールボックス・レジスタに書き込まれてS P U読み出しインバウンド・メールボックス・キューにデータ項目が置かれるまで、S P Uの実行をストールさせる。ストール状態を回避するために、このチャンネルに関連したチャンネル・カウントを読み出して、チャンネル読み出しを発行する前に、S P U読み出しインバウンド・メールボックス・キュー内にデータがあることを保証することができる。代わりに、S P Uインバウンド・メールボックス使用可能イベントを使用して、S P U読み出しインバウンド・メールボックス・キュー内のデータの使用可能性を知らせることもできる。

#### 【0184】

メールボックスが空の場合は、チャンネル・カウント (r c h c n t) は「0」の値を返す。r c h c n tの結果が非ゼロの場合は、メールボックスは、P P Eによって書き込まれたがS P Uによって読まれてはいない情報を含む。

#### 【0185】

S P U読み出しインバウンド・メールボックス・チャンネルのチャンネル・カウントは、特権的ソフトウェアによって「0」に初期化される。最大カウントは、実施に依存する。このチャンネルは、読み出しブロッキングである。

#### 【0186】

##### S P U信号通知ファシリティ

M F Cは、システム内の他のプロセッサおよび装置からのS P Uに対するバッファ完了フラグのように、信号を送るために使用されるS P U信号通知ファシリティを提供する。この信号通知ファシリティは、例えば、図2のM F Cレジスタユニット250内に提供される。

#### 【0187】

B P Aは、S P U信号通知1と、S P U信号通知2という、2つの独立した信号通知ファシリティを提供する。各ファシリティは、1つのレジスタと、1つのチャンネルからなる。すなわち、S P U信号通知1レジスタおよびS P U信号通知1チャンネルと、S P U信号通知2レジスタおよびS P U信号通知2チャンネルである。

#### 【0188】

信号は、S P Uが、信号送信コマンドのセットを、信号が送られるS P Uに関連した信号通知レジスタの有効アドレスと共に使用することによって、発行される。信号送信コマンドをサポートしていないP P Eおよび他の装置は、信号が送られるS P Uに関連したS P U信号通知レジスタへのM M I O書き込みを行うことによって、信号コマンドを送ることをシミュレートする。

#### 【0189】

各信号通知ファシリティは、1対1の信号方式環境において有用な上書きモードか、多対1の信号方式環境において有用な論理和モードのいずれかにプログラムできる。各チャンネルのモードは、S P U構成レジスタ内に設定される。

#### 【0190】

信号送信コマンド、または上書きモードにプログラムされたシグナリング・レジスタを対象とするM M I Oを実行すると、関連チャンネルの内容が、信号動作のデータに設定さ

れる。また、この実行は、対応チャンネル・カウントを「1」に設定する。論理和モードにおいて、信号動作のデータは、チャンネルの現在の内容との論理和が取られて、対応カウントは、「1」の値に設定される。

#### 【0191】

加えて、関連性のないロードを行う場合に、信号通知レジスタを画像の有効アドレスとして使用する。これらの場合に、S P U信号通知1レジスタは、64ビットの有効アドレスの上位32ビットを含み、S P U信号通知2レジスタは、最小桁の32ビットを含む。ソフトウェアは、関連性のないロード要求の適切な動作のために、上書きモードにおいてS P U信号通知ファシリティを有しなければならない。

#### 【0192】

##### S P Uシグナリング・チャンネル

S P Uシグナリング・チャンネルは、S P U信号通知ファシリティのP P E部分である。これらは、システム内の他のプロセッサおよび他の装置からの信号を読み出すのに使用される。シグナリング・チャンネルは、最大カウント「1」の読み出しブロッキングとして構成される。チャンネル読み出し (r d c h) 命令がこれらのチャンネルのうちの1つに送られ、かつ関連チャンネル・カウントが「1」の場合に、チャンネルの現在の内容および関連カウントは「0」に再設定される。チャンネル読み出し (r d c h) 命令は、これらのチャンネルのうちの1つに送られ、チャンネル・カウントは「0」である場合に、処理または動作がM M I O書き込みを関連レジスタに対して行うまで、S P Uはストールする。

#### 【0193】

##### S P U信号通知チャンネル

図38は、本発明の例示的な一実施形態に係るS P U信号通知チャンネルの詳細を示す。信号通知チャンネルは、S P U信号通知1または2チャンネルであってもよい。S P U信号通知チャンネルへ送られたチャンネル読み出し (r d c h) 命令は、信号制御ワード1の32ビットワードを返し、読み出す際に設定された任意のビットをアトミック的に再設定する。保留中の信号がない場合には、このチャンネルからの読み出しにより、S P Uは、信号が発行されるまでストールする。保留中の信号がない場合には、このチャンネルへ送られたチャンネル・カウント読み出し (r c h c n t) 命令は「0」を返す。読み出していない信号が保留中の場合には、「1」を返す。

#### 【0194】

特権的ソフトウェアは、このチャンネルについてのカウントを「0」の値に初期化する。このチャンネルは、最大カウント「1」で書き込みブロッキングイネーブルされる。

#### 【0195】

##### S P Uデクリメンタ

各S P Uは、32ビットのデクリメンタを含む。M F C制御レジスタ内でイネーブルであれば、「0」に設定されたM F C \_\_ C N T L [D h] が書き込まれる。S P Uデクリメンタは、チャンネル書き込み (w r c h) 命令がS P U書き込みデクリメンタチャンネルへ発行されると開始する。デクリメンタは、後述のプログラム・シーケンスに従って、またはM F C制御レジスタに「1」に設定されたM F C \_\_ C N T L [D h] が書き込まれる場合に停止する。デクリメンタの現在稼働中のステータスは、M F C制御レジスタ (すなわち、M F C \_\_ C N T L [D s]) において使用可能である。デクリメンタ・イベントは、デクリメンタを停止させるために保留させる必要はない。

#### 【0196】

デクリメンタの管理のために、2つのチャンネルが割り当てられる。1つは、デクリメンタ値を設定するためのもので、もう1つは、デクリメンタの現在の内容を読み出すためのものである。デクリメンタ・イベントは、最上位ビット (ビット0) が「0」から「1」へ変更すると生じる。

#### 【0197】

##### S P U書き込みデクリメンタ・チャンネル

10

20

30

40

50

図39は、本発明の例示的な一実施形態に係るSPU書き込みデクリメンタ・チャンネルの詳細を示す。SPU書き込みデクリメンタ・チャンネルは、デクリメンタの32ビット値をロードするために使用される。デクリメンタにロードされた値は、チャンネル書き込み(wrch)命令と、デクリメンタ・イベントとの間の経過時間を決定する。デクリメンタの最上位ビット(msb)が「0」から「1」変化すると、イベントが生じる。デクリメンタにロードされた値がmsbにおける「0」から「1」への変化を生じさせると、イベントは即座に知らされる。デクリメンタを「0」の値に設定すると、1つ分のデクリメンタ間隔の後のイベントという結果が生じる。

【0198】

デクリメンタの状態を適切に退避および復元するためには、デクリメンタは、デクリメンタ値を変更する前に停止されなければならない。以下のシーケンスは、新たなデクリメンタ値を設定するための手順の概略である。

【0199】

1. SPU書き込みイベント・マスク・チャンネルに書き込んで、デクリメンタ・イベントをディスエーブルにする。

【0200】

2. SPU書き込みイベント確認チャンネルに書き込んで任意の保留イベントを確認してデクリメンタを停止させる。デクリメンタが停止するのは、デクリメンタ・イベントがステップ1でディスエーブルされたからである。

【0201】

3. SPU書き込みデクリメンタチャンネルに書き込んで、新たなデクリメンタ・カウント値を設定する(注:デクリメンタが開始するのは、ステップ2がデクリメンタを停止していたからである)。

【0202】

4. SPU書き込みイベント・マスク・チャンネルに書き込んで、デクリメンタ・イベントをイネーブルにする。

【0203】

5. タイマが期限切れになるのを待つ。

【0204】

このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し(rchcnt)命令がこのチャンネルへ送られる度に、カウントは「1」として常に返される。

【0205】

#### SPU読み出しデクリメンタチャンネル

図40は、本発明の例示的な一実施形態に係るSPU読み出しデクリメンタ・チャンネルの詳細を示す。SPU読み出しデクリメンタ・チャンネルは、32ビットのデクリメンタの現在の値を読み出すために使用される。デクリメンタ・カウントの読み出しは、デクリメンタの精度に影響を与えない。デクリメンタの連続読み出しは、同じ値を返す。

【0206】

このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し(rchcnt)命令がこのチャンネルへ送られる度に、カウントは「1」として常に返される。

【0207】

#### SPU状態管理チャンネル

上記に加えて、SPU状態管理チャンネルが提供される。これらのSPU状態管理チャンネルは、SPU読み出し・マシン・ステータスチャンネルと、2つの割り込み関連状態チャンネルとを含む。割り込み関連状態チャンネルは、SPU書き込み状態退避/復元チャンネルと、SPU読み出し状態退避/復元チャンネルとを含む。

【0208】

図41は、本発明の例示的な一実施形態に係るSPU読み出し・マシン・ステータス・

チャンネルの詳細を示す。SPU読み出し・マシン・ステータス・チャンネルは、現在のSPUマシン・ステータス情報を含む。このチャンネルは、分離ステータスと、SPU割り込みステータスという2つのステータス・ビットを含む。この分離ステータスは、分離または非分離という、SPUの現在の動作状態を反映している。

#### 【0209】

SPU割り込みイネーブル・ステータスは、SPU割り込みイネーブルの現在の状態を反映している。イネーブルの場合、任意のイネーブルなSPUイベントがあれば、SPU割り込みが生成される。

#### 【0210】

このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し (rchcnt) 命令がこのチャンネルへ送られる度に、カウントは「1」として常に返される。

#### 【0211】

##### SPU書き込み状態退避／復元チャンネル

図42は、本発明の例示的な一実施形態に係るSPU書き込み状態退避／復元チャンネルの詳細を示す。このチャンネルへの書き込みにより、SPU内の状態退避／復元レジスタ0 (SRR0) の内容が更新される。このチャンネルへの書き込みは、典型的には、入れ子型の割り込みがサポートされる場合に、割り込み状態情報を復元するために使用される。

#### 【0212】

このチャンネルは、SPU割り込みがイネーブルの場合は書き込まれてはならない。それをおこなうと、SRR0の内容が不確かになる。このチャンネルの書き込み後であって、SRR0の内容に依存する命令の実行前に、同期命令のチャンネル形式が発行されなければならない。

#### 【0213】

このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し (rchcnt) 命令がこのチャンネルへ送られる度に、カウントは「1」として常に返される。

#### 【0214】

##### SPU読み出し状態退避／復元チャンネル

図43は、本発明の例示的な一実施形態に係るSPU読み出し状態退避／復元チャンネルの詳細を示す。このチャンネルの読み出しにより、SPU内の状態退避／復元レジスタ0 (SRR0) の内容が返される。このチャンネルへの読み出しは、典型的には、入れ子型の割り込みがサポートされる場合に、割り込み状態情報を退避するために使用される。

#### 【0215】

このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し (rchcnt) 命令がこのチャンネルへ送られる度に、カウントは「1」として常に返される。

#### 【0216】

##### SPUイベント・ファシリティ

図44および図45は、本発明の例示的な一実施形態に係るSPUイベント・ファシリティの論理表現を示すブロック図の例である。図44および図45に示すように、エッジトリガされたイベントが、SPU保留イベント・レジスタ2110内の対応ビットを「1」に設定する。SPU保留イベント・レジスタ2110内のイベントは、チャンネル命令を使用して、SPU書き込みイベント確認チャンネル2110内の対応ビットに「1」を書き込むことによって確認または再設定される。

#### 【0217】

SPU保留イベント・レジスタ (Pend\_Event) 2110は、内部レジスタである。SPU保留イベント・レジスタ2110は、SPUチャンネル・アクセス・ファシリティを使用して読み出すことができる。

10

20

30

40

50

## 【0218】

S P U読み出しイベント・ステータス・チャンネル2130をチャンネル読み出し (r d c h) 命令で読み出すと、S P U書き込みイベント・マスク・チャンネル2140内の値との論理積が取られたS P U保留イベント・レジスタの値を返す。この昨日は、S P Uプログラムに、イネーブルなイベントの上体のみを提供する一方で、S P U保留イベント・レジスタ2110は、特権的ソフトウェアに対して、発生したすべてのイベントを見ることができるようにしている。すべてのイベントに対するアクセスには、S P Eコンテキスト退避/復元動作が必要である。

## 【0219】

S P U読み出しイベント・ステータス・チャンネル2130の内容は、S P U書き込みイベント・マスク・チャンネル2140に新たな値が書き込まれる場合、または、新たなイベントがS P U保留イベント・レジスタ2110に記録される場合に、変化する。S P U読み出しイベント・ステータス・チャンネル2130内の「0」から「1」へのビットの変化によって、S P U読み出しイベント・ステータス・チャンネル・カウントが「1」インクリメントされる。また、カウントは、S P U書き込みイベント確認チャンネル2120に書き込みが送られた後にS P U読み出しイベント・ステータス・チャンネル2130内にまだイベントが設定されている場合にもインクリメントする。カウントは、S P U読み出しイベント・ステータス・チャンネル2130がチャンネル読み出し (r d c h) 命令を使用して読み出される場合に、「1」デクリメントされる。カウントは、「1」の値で飽和し、「0」の値を回ってデクリメントされない。S P U読み出しイベント・ステータス・チャンネル・カウントが非ゼロの場合、イネーブルであれば、割り込み状態がS P Uへ送られる。

## 【0220】

S P Uイベント・チャンネル

S P Uプログラムは、数多くのS P Uイベント・チャンネルを使用してイベントを監視してもよい。これらのS P Uイベント・チャンネルは、S P U読み出しイベント・ステータス・チャンネルと、S P U読み出しイベント・マスク・チャンネルと、S P U書き込みイベント確認チャンネルとを含む。S P U読み出しイベント・ステータス・チャンネルは、S P U書き込みイベント・マスク・チャンネル内のイネーブルなすべてのイベントのステータスを含む。S P U書き込みイベント確認チャンネルは、イベントのステータスを再設定するために使用され、これは、通常、イベントがS P Uプログラムによって処理または記録されたことを示す。イネーブルなイベントがない場合には、S P U読み出しイベント・ステータス・チャンネルからの読み出しにより、S P Uプログラムはストールする。

## 【0221】

個々のイベントがS P Uプログラムをストールするための同様の方法を有する場合、イベントが生じなければ、S P Uイベント・ファシリティは、ソフトウェアに対して、複数のイベントを探してS P Uプログラムに割り込みを生じさせるための方法を提供する。

## 【0222】

S P U読み出しイベント・ステータス・チャンネル

図46は、本発明の例示的な一実施形態に係るS P U読み出しイベント・ステータス・チャンネルの詳細を示す。S P U読み出しイベント・ステータス・チャンネルは、このチャンネルが読み出されたときにS P U書き込みイベント・マスク・チャンネルによってイネーブルされたすべてのイベントの現在のステータスを含む。S P U書き込みイベント・マスク・チャンネルが、イベントはクエリの一部ではないと指定する場合には、その対応位置は、報告されたステータス内の「0」である。

## 【0223】

「0」のチャンネル・カウントを有するS P U読み出しイベント・ステータス・チャンネルからの読み出しにより、S P Uストールという結果となつて、「イベント待ち」機能を提供する。チャンネル・カウント「1」でのこのチャンネルからの読み出しは、任意のイネーブルな保留イベントのステータスを返し、チャンネル・カウントを「0」に設定す

る。以下の条件の場合には、チャンネル・カウントは「1」に設定される。

【0224】

・イベントが発生し、SPU書き込みイベント・マスク・チャンネル内の対応マスクが「1」である場合。

【0225】

・SPU書き込みイベント・マスク・チャンネルに対して、SPU保留イベント・レジスタ内の「1」に対応するビット位置に「1」が書き込まれる場合。

【0226】

・SPU書き込みイベント確認チャンネルの書き込み後、イネーブルなイベントが保留中の場合。

【0227】

・特権的ソフトウェアが、SPUチャンネル・アクセス・ファシリティを使用して、チャンネル・カウントを「1」に設定する場合。

【0228】

イネーブルなイベントが発生しなかった場合、SPU読み出し・イベント・ステータス・チャンネルのチャンネル・カウント読み出し（rchcnt）命令は、ゼロを返す。SPU読み出しイベント・ステータス・チャンネルからイベント・ステータスを読み出す場合に、SPUがストールするのを回避するために、チャンネル・カウント読み出し（rchcnt）命令を使用することができる。

【0229】

特権的ソフトウェアは、SPU読み出し・イベント・ステータス・チャンネルについてのカウントを「0」に初期化する。チャンネル・カウントは、SPUチャンネル・アクセス・ファシリティ内のSPUチャンネル・カウント・レジスタを使用して初期化される。SPU割り込みがイネーブルの場合（SPU\_RdMachStat[IE]が「1」に設定）、非ゼロSPU読み出し・イベント・ステータス・チャンネル・カウントにより、SPUに対して割り込みを生じさせることになる。

【0230】

ソフトウェアは、以下の2つの場合に、疑似イベントを発生させることができる。

【0231】

1. イベントがSPU読み出しイベント・ステータス・チャンネル・カウントをインクリメントした後であって、SPU読み出しイベント・ステータス・チャンネルからイベント・ステータスを読み出す前に、ソフトウェアがイベントを確認またはマスクした場合。この場合には、SPU読み出しイベント・ステータス・チャンネルの読み出しは、イベントはもはや存在しないか、またはディスエーブルであることを示すデータを返す。

【0232】

2. SPU読み出しイベント・ステータス・チャンネルの読み出し前であって、イベントを確認する前に、イネーブルなイベントの割り込み条件をソフトウェアが再設定する場合（例えば、メールボックスからの読み出し）。この場合、イベント・ステータスレジスタの読み出しは、イベントを生成した条件はもはや存在しないがイベントはまだ保留中であることを示すデータを返す。この場合、イベントは依然として確認されなければならない。

【0233】

疑似イベントの生成を回避するために、イベントは以下のように処理されなくてはならない。

【0234】

・SPU読み出しイベント・ステータス・チャンネルを読み出す。

【0235】

・処理すべきすべてのイベントについて、SPU書き込みイベント確認チャンネルに対応ビットを書き込むことによって、イベントを確認する。

【0236】

10

20

30

40

50



・イベントを処理する（例えば、メールボックスを読む、タイマを再設定または停止させる、もしくは信号通知レジスタについて読み出す）

【0237】

S P U書き込みイベント・マスク・チャンネル

図47は、本発明の例示的な一実施形態に係るS P U書き込みイベント・マスク・チャンネルの詳細を示す。S P U書き込みイベント・マスク・チャンネルは、S P U読み出しイベント・ステータス・チャンネルの状態に影響を与える保留イベントを選択する。このチャンネルの内容は、後続のチャンネル書き込みまたはS P Uチャンネル・アクセスが生じるまで、保持される。このチャンネルの現在の内容は、S P U読み出しイベント・マスク・チャンネルを読み出すことによって、アクセスできる。

10

【0238】

すべてのイベントは、S P Uイベントマスク設定に関わらず、S P U保留イベント・レジスタに記録される。チャンネル書き込み(w r c h)命令によってS P U書き込みイベント確認チャンネルにクリアされるまで、または、特権的ソフトウェアがS P Uチャンネル・アクセス・ファシリティを使用してS P U保留イベント・レジスタに新たな値をロードするまでは、イベントは保留のままである。保留イベントは、ディスエーブルになった場合もクリアされる。

【0239】

保留イベントは、ディスエーブルされた後にクリアされるが、これはS P U読み出しイベント・ステータスチャンネルに反映されていない。保留イベントをイネーブルすると、イネーブルであれば、S P U読み出しイベント・ステータス・チャンネルの更新およびS P U割り込みという結果となる。

20

【0240】

このチャンネルは、非ブロッキング・チャンネルであり、関連カウントを有しない。このチャンネルのチャンネル・カウント読み出し(r c h c n t)命令は、常に「1」を返す。

【0241】

S P U読み出しイベント・マスク・チャンネル

図48は、本発明の例示的な一実施形態に係るS P U読み出しイベント・マスク・チャンネルの詳細を示す。S P U読み出しイベント・マスク・チャンネルは、イベント・ステータス・マスクの現在の値を読み出すために使用される。このチャンネルを読み出すと、S P U書き込みイベント・マスク・チャンネルによって書き込まれた最終データを常に返す。このチャンネルは、イベント・ステータス・マスクのソフトウェア・シャドウ・コピーを回避するため、およびS P Eコンテキスト退避/復元動作のために、使用することができる。このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し(r c h c n t)命令をこのチャンネルへ送る度に、カウントは常に「1」として返される。

30

【0242】

S P U書き込みイベント確認チャンネル

図49は、本発明の例示的な一実施形態に係るS P U書き込みイベント確認チャンネルの詳細を示す。特定のイベントビットを設定した状態でのS P U書き込みイベント確認チャンネルへの書き込みは、対応のイベントがソフトウェアによってサービス提供中であることを確認する。確認されたイベントは、再設定および再サンプリングされる。報告はされたが確認はされていないイベントは、確認されるまで、またはS P Uチャンネル・アクセス・ファシリティを使用して特権的ソフトウェアによってクリアされるまで、報告が継続される。

40

【0243】

ディスエーブルなイベントは、S P U読み出しイベント・ステータス・チャンネルにおいては報告されないが、S P U書き込みイベント確認チャンネル内の対応ビットに「1」が書き込まれることによってクリアされるまで、保留のままとなる。ディスエーブルなイ

50

イベントを確認すると、当該イベントは、報告はされなかったがクリアされる。生じつ前にイベントをクリアすると、ソフトウェアによって引き起こされるデッドロックという結果となる。

#### 【0244】

このチャンネルは、非ブロッキングであり、関連カウントを有しない。チャンネル・カウント読み出し (`rchcnt`) 命令がこのチャンネルへ送られる度に、カウントは「1」として常に返される。

#### 【0245】

##### S P U イベント

ハードウェアは、適切なチャンネル・カウント、デクリメンタカウント、または S P U チャンネル・アクセス動作を検出することによってイベントを判断する。数多くの互いに異なる種類のイベントが、上述の B P A によってサポートされている。例えば、M F C タグ・グループ・ステータス更新イベントは、M F C 読み出しタグ・グループ・ステータス・チャンネルについてのカウントが 0 から非ゼロ値へ変化する場合に設定される。M F C ・D M A リスト・コマンド・ストール／通知イベントは、M F C 読み出しリスト・ストール／通知タグ・ステータス・チャンネルについてのカウントが 0 から非ゼロ値へ変化する場合に設定される。M F C ・S P U コマンド・キュー使用可能イベントは、キューイングされた M F C コマンド操作符号レジスタについてのカウントが 0 (満杯) から非ゼロ値 (満杯でない) へ変化する場合に設定される。S P U インバウンド・メールボックス使用可能イベントは、S P U 読み出しインバウンド・メールボックス・チャンネルが 0 から非ゼロ値へ変化する場合に設定される。

#### 【0246】

同様に、S P U デクリメンタ・イベントは、デクリメンタ・カウントの最上位ビットが 0 から 1 に変化する場合に設定される。デクリメンタにロードされた値が最上位ビットにおいて 0 から 1 に変化する場合には、イベントが即座に知らされる。デクリメンタ値を 0 に設定すると、1 つ分のデクリメンタ間隔の後のイベントという結果が生じる。

#### 【0247】

さらに、S P U アウトバウンド・メールボックス使用可能イベントは、S P U 書き込みアウトバウンド割り込みメールボックス・チャンネル・カウントが 0 から非ゼロの値に変化した場合に設定される。S P U 信号通知 1 または 2 使用可能イベントは、対応する S P U 信号通知チャンネルが 0 から非ゼロの値に変化した場合に設定される。ロック・ライン予約イベントは、アトミック予約が失われた場合に設定される (後述の「ロック・ライン予約喪失イベント」の項を参照)。特権的アテンション・イベントは、S P U 特権的制御レジスタがアテンション・イベント要求ビットが 1 に設定されて書き込まれた場合に設定される (後述の「特権的アテンション・イベント」の項を参照)。マルチソース同期イベントは、M F C 書き込みマルチソース同期要求チャンネルが 0 から 1 の値に変化した場合に設定される。これらのイベントをさらに詳細に説明する。

#### 【0248】

##### M F C タグ・グループ・ステータス更新イベント

M F C タグ・グループ・ステータス更新イベントは、1 つのタグ・グループまたは複数のグループが完了したことと、M F C 読み出しタグ・グループ・ステータスチャンネルが更新されたこととを S P U プログラムに通知するために使用され、S P U をストールせずに読み出すことができる (上記「M F C タグ・グループ・ステータス・チャンネル」の項を参照)。M F C 読み出しタグ・グループ・ステータスチャンネルが「0」から「1」へ変化した場合に生じる。イベントが生じたことにより、`Pend_Event[Tg]` が「1」へ設定される。イベントがイネーブル (すなわち、`S P U_RdEventStat[Tg]` が「1」に設定される) の場合は、S P U イベント・ステータス・チャンネルについてのカウントが「1」に設定される。

#### 【0249】

チャンネル書き込み (`wrch`) が S P U 保留イベント・レジスタへ発行された場合、

または、特権的ソフトウェアが、対応ビットが「0」に設定されてSPUチャンネルアクセス・ファシリティを使用してSPU保留イベント・レジスタを更新する場合は、Pend\_Event[Tg]ビットが0に設定される。このイベントは、タグ・グループまたは複数のグループに対する任意のコマンドを発行する前にクリアされなければならない。

#### 【0250】

##### MFC・DMAリスト・コマンド・ストール／通知イベント

MFC・DMAリスト・コマンド・ストール／通知イベントは、MFC・DMAリスト・コマンド内のリスト要素が完了して、MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルが更新されたことをSPUプログラムに通知するために使用され、SPUをストールせずに読み出すことができる。このイベントは、MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルが「0」から「1」へ変化したときに生じる。

#### 【0251】

ストール／通知フラグが設定されているすべてのリスト要素の転送と、MFC・DMAリスト・コマンド内のすべての以前のリスト要素についての転送とが関連SPEに対して完了した場合に、カウンタが「1」に設定される。このイベントが生じると、Pend\_Event[Sn]は「1」に設定される。このイベントがイネーブルされると（すなわち、PU\_RdEventStat[Sn]が「1」に設定されると）、SPU読み出しイベント・ステータス・チャンネルは「1」に設定される。タグ・ビットが「1」に設定されたSPU書き込みイベント確認チャンネルへチャンネル書き込み(wrch)が発行されるか、対応ビットが「0」に設定されたSPUチャンネル・アクセス・ファシリティを使用して特権的ソフトウェアがSPU保留イベント・レジスタを更新する場合に、Pend\_Event[Sn]ビットは「0」に設定される。

#### 【0252】

MFC・DMAリスト・コマンド・ストール／通知イベントの処理のための手順を図56に概略的に示す。図56に示すように、手順は、チャンネル読み出し(rdch)命令をSPU読み出しイベント・マスク・チャンネルに対して実行して、データを「マスク」に退避することによって開始する(ステップ2310)。イベントは、SPU\_WriteEventMask[Sn]が「0」に設定されたSPU書き込みイベント・マスク・チャンネルに、チャンネル書き込み命令を発行することによってマスクされる(ステップ2320)。イベントは、SPU\_WriteEventAck[Sn]が1に設定されたSPU書き込みイベント確認チャンネルに対してチャンネル書き込み(wrch)命令を実行することによって確認される(ステップ2330)。

#### 【0253】

その後、読み出しチャンネル(rdch)命令を、MFC読み出しリスト・ストール／通知タグ・ステータス・チャンネルMFC\_StallStat[g\_n]へ送る(ステップ2340)。返された情報を使用して、どのタグ・グループまたは複数のタグ・グループがストール／通知状態のDMAリスト要素を有しているかを判断する(ステップ2350)。その後、ストールされたDMAリスト要素を有するタグ・グループ毎にアプリケーション別の動作を実行する(ステップ2360)。

#### 【0254】

ストールされた各DMAリスト・コマンドは、チャンネル書き込み(wrch)命令をリスト・ストール／通知タグ確認チャンネルMFC\_StallAck[MFC\_Tag]に発行することによって、確認および再開される(ステップ2370)。当該チャンネルでは、与えられたMFCタグは、再開すべきタグ・グループの符号化されたタグIDである。その後、DMAリスト・ストール／通知ハンドラを終了する(ステップ2380)。ソフトウェアは、MFC\_StallStat[g\_n]チャンネルにおいて示されているすべてのストールされたタグ・グループを確認するのではなく、第2のストール／通知イベントは未確認のタグ・グループには生じないことに注意すべきである。

#### 【0255】

チャンネル書き込み (w r c h) 命令を S P U \_ W r E v e n t M a s k [ m a s k ] を有する S P U 書き込みイベント・マスク・チャンネルに発行することによって、マスクを復元する (ステップ 2 3 9 0)。その後、一般イベント・ハンドラを終了する (ステップ 2 3 9 5)。

#### 【0256】

DMA リストがストール/通知フラグが設定された複数のリストを含み、または、タグ・グループがストール/通知フラグが設定された要素でキューイングされた複数の DMA リスト・コマンドを含む場合、ならびにその両方の場合には、DMA リスト・コマンドがタグ・グループにキューイングされる前に、アプリケーション・ソフトウェアは、タグ・グループ別のストール・カウンタを 0 に初期化する。加えて、複数の DMA リスト・コマンドがストール/通知要素を有するタグ・グループのためにキューイングされている場合には、タグ別のフェンス、バリア、またはコマンド・バリアで順序付けが実現される。ストール/通知ステータスがタグ・グループに示される度に、対応カウンタがインクリメントされる。その後、アプリケーション・ソフトウェアはこのカウンタを使用して、リスト内のどの点でストールが生じるのかを判断する。

#### 【0257】

アプリケーション・ソフトウェアは、ストール/通知を使用して、動的に変化する条件のためにストールしたリスト要素に続くリスト要素アドレスと、転送サイズとを更新する。ストールされたリスト要素の後のリスト要素は、それらの転送サイズを 0 に設定することによって、スキップすることができる。しかしながら、キューイングされた DMA リストコマンド内のリスト要素の数は、変更することができない。

#### 【0258】

##### MFC・SPU コマンド・キュー使用可能イベント

MFC・SPU コマンド・キュー使用可能イベントは、MFC・SPU コマンド・キュー使用可能イベント内のエントリが使用可能であり、MFC コマンド操作符号チャンネルが SPU をストールせずに書き込みできることを SPU プログラムに通知するために使用される。このイベントは、MFC マンド操作符号チャンネルについてのチャンネルカウンタが「0」(フル)から非ゼロ(非フル)の値に変化することで生じる。

#### 【0259】

MFC・DMA コマンド・キュー内の 1 つ以上の MFC・DMA コマンドが完了した場合に、カウンタが「1」に設定される。このイベントが生じると、P e n d \_ E v e n t [ Q v ] は「1」に設定される。このイベントがイネーブルされると(すなわち、P U \_ R d E v e n t M a s k [ Q v ] が「1」に設定されると)、S P U \_ R d E v e n t S t a t [ Q v ] が「1」に設定され、S P U 読み出しイベント・ステータス・チャンネルについてのカウンタが「1」に設定される。チャンネル書き込み (w r c h) が S P U 書き込みイベント確認チャンネルに発行される(すなわち、S P U \_ W r E v e n t A c k [ Q v ]) 場合、または、特権的ソフトウェアが、対応ビットが「0」に設定された S P U チャンネル・アクセス・ファシリティを使用して S P U 保留イベント・レジスタを更新する場合に、P e n d \_ E v e n t [ Q v ] ビットは「0」に設定される。

#### 【0260】

MFC・SPU コマンド・キュー使用可能イベントを処理する手順を図 5 7 に概略的に示す。図 5 7 に示すように、手順は、チャンネル読み出し (r d c h) 命令を S P U 読み出しイベント・マスク・チャンネルへ送って、データを「マスク」に退避することによって開始する (ステップ 2 4 1 0)。イベントは、S P U \_ W r E v e n t M a s k [ Q v ] が「0」に設定された S P U 書き込みイベント・マスク・チャンネルに、チャンネル書き込み命令を発行することによってマスクされる (ステップ 2 4 2 0)。イベントは、S P U \_ W r E v e n t A c k [ Q v ] が 1 に設定された S P U 書き込みイベント確認チャンネルに対してチャンネル書き込み (w r c h) 命令を実行することによって確認される (ステップ 2 4 3 0)。

#### 【0261】

チャンネル・カウント読み出し (rchcnt) 命令をMFCコマンド操作符号チャンネル (MFC\_CMD) に発行することによって、チャンネル・カウントを取得する (ステップ2440)。チャンネル・カウントが「0」かどうかについての判断がなされる (ステップ2450)。もし「0」でなければ、MFCコマンドキューへDMAコマンドをエンキューする (ステップ2460)。その後、キューに追加のコマンドが残っているかどうかについての判断がなされる (ステップ2470)。もし残っていれば、処理はステップ2430へ戻る。追加のコマンドが残っている場合、またはチャンネル・カウントが「0」の場合は、SPUコマンド・キュー・ハンドラを終了する (ステップ2490)。その後、一般イベント・ハンドラを終了する (ステップ2495)。

【0262】

#### SPUインバウンド・メールボックス使用可能イベント

SPUインバウンド・メールボックス使用可能イベントは、PPEまたは他の装置が、空のSPUメールボックスに書き込みを行ったこと、およびSPU読み出しインバウンド・メールボックス・チャンネルがSPUをストールすることなく読み出されたことを、SPUプログラムに通知するために使用される。このイベントがイネーブルされる (すなわち、SPU\_RdEventStat[Mb] が「1」であれば)、SPU読み出しイベント・ステータス・チャンネルは「1」に設定される。

【0263】

このイベントは、SPU読み出しインバウンド・メールボックス・チャンネルについてのカウントが「0」(空)から非ゼロ(空でない)値に変化した場合に生じる。このイベントが生じると、Pend\_Event[Mb] は「1」に設定される。チャンネル書き込み (wrch) がSPU書き込みイベント確認チャンネルに発行される (すなわち、SPU\_WrEventAck[Mb] が「1」に設定される) か、または、特権的ソフトウェアが、対応のビットが「0」に設定されたSPUチャンネル・アクセス・ファシリティを使用してSPU保留イベント・レジスタを更新する場合には、Pend\_Event[Mb] ビットは「0」に設定される。

【0264】

SPUインバウンド・メールボックス使用可能イベントの処理のための手順を図58に概略的に示す。図58に示すように、手順は、チャンネル読み出し (rdch) 命令をSPU読み出しイベント・マスク・チャンネルに対して送って、データを「マスク」に退避することによって開始する (ステップ2510)。その後、イベントは、SPU\_WrEvent[Masking] が「0」に設定されたSPU書き込みイベント・マスク・チャンネルに、チャンネル書き込み命令を発行することによってマスクされる (ステップ2520)。イベントは、SPU\_WrEventAck[Sn] が1に設定されたSPU書き込みイベント確認チャンネルに対してチャンネル書き込み (wrch) 命令を実行することによって確認される (ステップ2530)。

【0265】

チャンネル・カウント読み出し (rchcnt) 命令をSPU読み出しインバウンド・メールボックス・チャンネルに発行することによって、チャンネル・カウントを取得する (ステップ2540)。チャンネル・カウントが「0」かどうかについての判断がなされる (ステップ2550)。もし「0」でなければ、チャンネル読み出し (rdch) 命令をSPU読み出しインバウンド・メールボックス・チャンネル (SPU\_RdInMbox) に発行することによって、次のメールボックスデータエントリを読み出す (ステップ2560)。その後、手順はステップ2530へ戻る。

【0266】

チャンネル・カウントが「0」の場合は、SPUインバウンド・メールボックス・ハンドラを終了する (ステップ2570)。チャンネル書き込み (wrch) 命令をSPU\_WrEventMask[mask] を有するSPU書き込みイベント・マスク・チャンネルへ発行することによって、「マスク」を復元する (ステップ2580)。その後、一般イベント・ハンドラを終了する (ステップ2590)。

10

20

30

40

50

## 【0267】

S P Uデクリメンタ・イベント

S P Uデクリメンタ・イベントは、デクリメンタが「0」に到達したことをS P Uに通知するために使用される。イベントがイネーブル（すなわち、S P U\_\_RdEventStat [Tm] が「1」に設定されている）であり、S P U読み出しイベント・ステータス・チャンネルについてのカウンタが「1」に設定されている場合に、デクリメンタの最上位ビットが「0」から「1」（負）の値に変化すると、このイベントが生じる。このイベントが生じると、Pend\_\_Event [Tm] は「1」に設定される。チャンネル書き込み（w r c h）がS P U書き込みイベント確認チャンネルに発行される（S P U\_\_WrEventAck [Tm] が「1」に設定される）か、または、特権的ソフトウェアが、対応のビットが「0」に設定されたS P Uチャンネル・アクセス・ファシリティを使用してS P U保留イベント・レジスタを更新する場合には、Pend\_\_Event [Tm] ビットは「0」に設定される。

10

## 【0268】

S P Uデクリメンタ・イベントの処理のための手順を図59に概略的に示す。図59に示すように、手順は、チャンネル読み出し（r d c h）命令をS P U読み出しイベント・マスク・チャンネルに対して実行して、データを「マスク」に退避することによって開始する（ステップ2610）。イベントは、S P U\_\_WrEventMask [Tm] が「0」に設定されたS P U書き込みイベント・マスク・チャンネルに、チャンネル書き込み命令（w r c h）を発行することによってマスクされる（ステップ2620）。イベントは、（S P U\_\_WrEventAck [Tm] が1に設定された）S P U書き込みイベント確認チャンネルに対してチャンネル書き込み（w r c h）命令を発行することによって確認される（ステップ2630）。

20

## 【0269】

チャンネル読み出し（r d c h）命令をS P U読み出しデクリメンタ・チャンネルに発行することによって、デクリメンタ値を読み出す（ステップ2640）。この値が負の場合は、所望の期間から追加の時間がどの程度経過したかを判断するのに使用できる。新たなタイマ・イベントを所望するかどうかについての判断が行われる（ステップ2650）。新たなタイマ・イベントを所望する場合には、新たなデクリメンタ値をS P U書き込みデクリメンタ・チャンネルへ書き込む（ステップ2660）。その後、または新たなタイマ・イベントを所望しない場合は、S P Uデクリメンタ・イベント・ハンドラを終了する（ステップ2670）。チャンネル書き込み（w r c h）命令をS P U\_\_WrEventMask [mask] を有するS P U書き込みイベント・マスク・チャンネルへ発行することによって、「マスク」を復元する（ステップ2680）。その後、一般イベント・ハンドラを終了する（ステップ2690）。

30

## 【0270】

S P Uアウトバウンド割り込みメールボックス使用可能イベント

S P Uアウトバウンド割り込みメールボックス使用可能イベントは、P P Eまたは他の装置が完全なS P Uアウトバウンド割り込みメールボックス・レジスタを読み出したことと、S P U書き込みアウトバウンド割り込みメールボックス・チャンネルがS P Uのストールなしで書き込み可能であることをS P Uプログラムに通知するために使用される。このイベントがイネーブル（すなわち、S P U\_\_RdEventStat [Me] が「1」に設定されている）である場合であって、S P U読み出しイベント・ステータス・チャンネルについてのカウンタが「1」に設定されている場合に、S P U書き込みアウトバウンド割り込みメールボックスチャンネルが「0」（フル）から非ゼロ（フルでない）値に変化すると、S P U読み出しイベント・ステータス・チャンネルについてのカウンタは「1」に設定される。このイベントは、Pend\_\_Event [Me] を「1」に設定する。チャンネル書き込み（w r c h）が、Meビットが「1」に設定された（すなわち、S P U\_\_WrEventAck [Me] が1に設定された）S P U書き込みイベント確認チャンネルに発行される場合、または、特権的ソフトウェアが、対応ビットが「0」に設定

40

50

されたSPUチャンネル・アクセス・ファシリティを使用してSPU保留イベント・レジスタを更新する場合に、Pend\_\_Event [Me] ビットは「0」に設定される。

#### 【0271】

SPUアウトバウンド割り込みメールボックス使用可能イベントを処理する手順を図60に概略的に示す。図60に示すように、手順は、チャンネル読み出し (rdch) 命令をSPU読み出しイベント・マスク・チャンネルへ送って、データを「マスク」に退避することによって開始する (ステップ2710)。イベントは、SPU\_\_WrEventMask [Me] が「0」に設定されたSPU書き込みイベント・マスク・チャンネルに、チャンネル書き込み (wrch) 命令を発行することによってマスクされる (ステップ2720)。イベントは、SPU\_\_WrEventAck [Me] が1に設定されたSPU書き込みイベント確認チャンネルに対してチャンネル書き込み (wrch) 命令を実行することによって確認される (ステップ2730)。

#### 【0272】

チャンネル・カウント読み出し (rchcnt) 命令をSPU書き込みアウトバウンド割り込みメールボックス・チャンネルに発行することによって、チャンネル・カウントを取得する (ステップ2740)。チャンネル・カウントが「0」かどうかについての判断がなされる (ステップ2750)。もしチャンネル・カウントが「0」でなければ、チャンネル書き込み (wrch) 命令をSPU書き込みアウトバウンド割り込みメールボックス・チャンネルへ発行することによって、新たなメールボックス・データ・エントリが書き込まれる (ステップ2760)。チャンネル・カウントが「0」の場合は、チャンネル・カウントが再び読み出される (ステップ2740)。その後、SPUアウトバウンド割り込みメールボックス使用可能ハンドラを終了する (ステップ2770)。チャンネル書き込み (wrch) 命令をSPU\_\_WrEventMask [mask] を有するSPU書き込みイベント・マスク・チャンネルへ発行することによって、「マスク」を復元する (ステップ2780)。その後、一般イベント・ハンドラを終了する (ステップ2795)。

#### 【0273】

##### SPUアウトバウンド・メールボックス使用可能イベント

SPUアウトバウンド・メールボックス使用可能イベントは、プロセッサまたは他の装置のいずれかが完全なSPUアウトバウンド・メールボックス・レジスタから読み出したことと、SPU書き込みアウトバウンド・メールボックス・チャンネルがSPUのストールなしで書き込み可能であることとをSPUプログラムに通知するために使用される。このイベントは、SPU書き込みアウトバウンド・メールボックス・チャンネルのチャンネル・カウントが「0」 (フル) から非ゼロ (フルでない) 値に変化すると生じる。このイベントが生じると、Pend\_\_Event [Le] は「1」に設定される。このイベントがイネーブルされると (すなわち、SPU\_\_RdEventStat [Le] が「1」に設定されると)、SPU読み出しイベント・ステータス・チャンネルについてのカウントは「1」に設定される。チャンネル書き込み (wrch) が、Leビットが「1」に設定された (すなわち、SPU\_\_WrEventAck [Le] が1に設定された) SPU書き込みイベント確認チャンネルに発行されるか、または、特権的ソフトウェアが、対応のビットが「0」に設定されたSPUチャンネル・アクセス・ファシリティを使用してSPU保留イベント・レジスタを更新する場合には、Pend\_\_Event [Le] ビットは「0」に設定される。

#### 【0274】

SPUアウトバウンド・メールボックス使用可能イベントを処理する手順を図61に概略的に示す。図61に示すように、手順は、チャンネル読み出し (rdch) 命令をSPU読み出しイベント・マスク・チャンネルへ送って、データを「マスク」に退避することによって開始する (ステップ2810)。イベントは、SPU\_\_WrEventMask [Le] が「0」に設定されたSPU書き込みイベント・マスク・チャンネルに、チャンネル書き込み (wrch) 命令を発行することによってマスクされる (ステップ2820

）。イベントは、SPU\_WrEventAck [Le] が「1」に設定されたSPU書き込みイベント確認チャンネルに対してチャンネル書き込み (wrc h) 命令を実行することによって確認される (ステップ2830)。

#### 【0275】

チャンネル・カウント読み出し (rch cnt) 命令をSPU書き込みアウトバウンドメールボックス・チャンネルに発行することによって、チャンネル・カウントを取得する (ステップ2840)。チャンネル・カウントが「0」かどうかについての判断がなされる (ステップ2850)。もしチャンネル・カウントが「0」でなければ、チャンネル書き込み (wrc h) 命令をSPU書き込みアウトバウンド・メールボックス・チャンネルへ発行することによって、新たなメールボックス・データ・エントリが書き込まれる (ステップ2860)。チャンネル・カウントが「0」の場合は、チャンネル・カウントが再び読み出される (ステップ2840)。その後、SPUアウトバウンド・メールボックス使用可能ハンドラを終了する (ステップ2870)。チャンネル書き込み (wrc h) 命令をSPU\_WrEventMask [mask] を有するSPU書き込みイベント・マスク・チャンネルへ発行することによって、「マスク」を復元する (ステップ2880)。その後、一般イベント・ハンドラを終了する (ステップ2895)。

#### 【0276】

##### SPU信号通知2使用可能イベント

SPU信号通知2使用可能イベントは、他のプロセッサまたは装置が空のSPU信号通知2レジスタに書き込みを行ったことと、SPU信号通知2チャンネルがSPUのストールなしで読み出し可能であることをSPUプログラムに通知するために使用される。このイベントは、SPU信号通知2チャンネルについてのチャンネル・カウントが「0」(空)から「1」(有効)の値へ変化すると生じる。このイベントがイネーブル(すなわち、SPU\_RdEventStat [S2] が「1」に設定されている)である場合であって、SPU読み出しイベント・ステータス・チャンネルについてのカウントが「1」に設定されている場合に、このイベントは、Pend\_Event [S2] ビットを「1」に設定する。チャンネル書き込み (wrc h) が、S2ビットが「1」に設定された (SPU\_WrEventAck [S2]) SPU書き込みイベント確認チャンネルに発行される場合、または、特権的ソフトウェアが、対応ビットが「0」に設定されたSPUチャンネル・アクセス・ファシリティを使用してSPU保留イベント・レジスタを更新する場合に、Pend\_Event [S2] ビットは「0」に設定される。

#### 【0277】

SPU信号通知2使用可能イベントを処理する手順を図62に概略的に示す。図62に示すように、手順は、チャンネル読み出し (rd ch) 命令をSPU読み出しイベント・マスク・チャンネルへ送って、データを「マスク」に退避することによって開始する (ステップ2910)。イベントは、SPU\_WrEventMask [S2] が「0」に設定されたSPU書き込みイベント・マスク・チャンネルに、チャンネル書き込み (wrc h) 命令を発行することによってマスクされる (ステップ2920)。イベントは、SPU\_WrEventAck [S2] が1に設定されたSPU書き込みイベント確認チャンネルに対してチャンネル書き込み (wrc h) 命令を実行することによって確認される (ステップ2930)。

#### 【0278】

チャンネル・カウント読み出し (rch cnt) 命令をSPU信号通知2チャンネルに発行することによって、チャンネル・カウントを取得する (ステップ2940)。チャンネル・カウントが「0」かどうかについての判断がなされる (ステップ2950)。もしチャンネル・カウントが「0」でなければ、チャンネル読み出し命令をSPU信号通知2チャンネルへ発行することによって、信号データを読み出す (ステップ2960)。その後、またはチャンネル・カウントが「0」の場合は、信号通知2ハンドラを終了する (ステップ2970)。チャンネル書き込み (wrc h) 命令をSPU\_WrEventMask [mask] を有するSPU書き込みイベント・マスク・チャンネルへ発行すること



によって、「マスク」を復元する（ステップ2980）。その後、一般イベント・ハンドラを終了する（ステップ2990）。

【0279】

#### S P U 信号通知 1 使用可能イベント

S P U 信号通知 1 使用可能イベントは、他のプロセッサまたは装置が空の S P U 信号通知 1 レジスタに書き込みを行ったことと、S P U 信号通知 1 チャンネルが S P U のストールなしで読み出し可能であることを S P U プログラムに通知するために使用される。このイベントは、S P U 信号通知 1 チャンネルについてのチャンネル・カウン트가「0」（空）から「1」（有効）の値へ変化すると生じる。このイベントがイネーブル（すなわち、S P U \_ R d E v e n t S t a t [ S 1 ] が「1」に設定されている）である場合であって、S P U 読み出しイベント・ステータス・チャンネルについてのカウン트가「1」に設定されている場合に、このイベントは、P e n d \_ E v e n t [ S 1 ] を「1」に設定する。チャンネル書き込み（w r c h）が、S 1 ビットが「1」に設定された（S P U \_ W r E v e n t A c k [ S 1 ]）S P U 書き込みイベント確認チャンネルに発行される場合、または、特権的ソフトウェアが、対応ビットが「0」に設定された S P U チャンネル・アクセス・ファシリティを使用して S P U 保留イベント・レジスタを更新する場合に、P e n d \_ E v e n t [ S 1 ] ビットは「0」に設定される。

【0280】

S P U 信号通知 1 使用可能イベントを処理する手順を図 6 3 に概略的に示す。図 6 3 に示すように、手順は、チャンネル読み出し（r d c h）命令を S P U 読み出しイベント・マスク・チャンネルへ送って、データを「マスク」に退避することによって開始する（ステップ3010）。イベントは、S P U \_ W r E v e n t M a s k [ S 1 ] が「0」に設定された S P U 書き込みイベント・マスク・チャンネルに、チャンネル書き込み（w r c h）命令を発行することによってマスクされる（ステップ3020）。イベントは、S P U \_ W r E v e n t A c k [ S 1 ] が 1 に設定された S P U 書き込みイベント確認チャンネルに対してチャンネル書き込み（w r c h）命令を実行することによって確認される（ステップ3030）。

【0281】

チャンネル・カウン読み出し（r c h c n t）命令を S P U 信号通知 1 チャンネルに発行することによって、チャンネル・カウンを取得する（ステップ3040）。チャンネル・カウンが「0」かどうかについての判断がなされる（ステップ3050）。

【0282】

もしチャンネル・カウンが「0」でなければ、チャンネル読み出し命令を S P U 信号通知 1 チャンネルへ発行することによって、信号データを読み出す（ステップ3060）。その後、またはチャンネル・カウンが「0」の場合は、信号通知 1 ハンドラを終了する（ステップ3070）。チャンネル書き込み（w r c h）命令を S P U \_ W r E v e n t M a s k [ m a s k ] を有する S P U 書き込みイベント・マスク・チャンネルへ発行することによって、「マスク」を復元する（ステップ3080）。その後、一般イベント・ハンドラを終了する（ステップ3090）。

【0283】

#### ロック・ライン予約喪失イベント

ロック・ライン予約喪失イベントは、キャッシュ・ライン上の予約の喪失が生じたバス・アクションについて S P U プログラムに通知するために使用される。予約は、S P U プログラムによって、g e t \_ l o c k \_ l i n e \_ a n d \_ r e s e r v e ( g e t l l a r ) コマンドを発行することによって取得される。予約は、他のプロセッサまたは装置がキャッシュ・ラインを予約で修正する場合に失われる。

【0284】

また、予約は、特権的ソフトウェアが M F C アトミック・フラッシュ・レジスタにフラッシュ・ビットを書き込む（M F C \_ A t o m i c \_ F l u s h [ F ] が「1」に設定される）場合に失われる場合がある。予約が失われると、イベントが生じる。このイベント

が生じると、Pend\_Event [Lr] は「1」に設定される。このイベントがイネーブルされると（すなわち、SPU\_ReadEventStat [Lr] が「1」に設定されると）、SPU読み出しイベント・ステータス・チャンネルについてのカウンタは「1」に設定される。チャンネル書き込み（wrch）が、Lrビットが「1」に設定された（SPU\_WriteEventAck [Lr]）SPU書き込みイベント確認チャンネルに発行されるか、または、特権的ソフトウェアが、対応のビットが「0」に設定されたSPUチャンネル・アクセス・ファシリティを使用してSPU保留イベント・レジスタを更新する場合には、Pend\_Event [Lr] ビットは「0」に設定される。

#### 【0285】

ロック・ライン予約喪失イベントを処理する手順を図64に概略的に示す。図64に示すように、手順は、チャンネル読み出し（rdch）命令をSPU読み出しイベント・マスク・チャンネルへ発行して、データを「マスク」に退避することによって開始する（ステップ3110）。イベントは、SPU\_WriteEventMask [Lr] が「0」に設定されたSPU書き込みイベント・マスク・チャンネルに、チャンネル書き込み（wrch）命令を発行することによってマスクされる（ステップ3120）。イベントは、SPU\_WriteEventAck [Lr] が「1」に設定されたSPU書き込みイベント確認チャンネルに対してチャンネル書き込み（wrch）命令を実行することによって確認される（ステップ3130）。

#### 【0286】

ロック・ライン領域のデータのシステム修正に応じて、アプリケーション別の機能を実行する（ステップ3140）。これは、通常、メモリ内のソフトウェア構造をチェックして、ロック・ラインがまだ監視されているかを判断することによって開始される。まだ「待っている」状態であれば、次のステップとしては、典型的には、getllarコマンドを修正されたのと同じロック・ライン領域へ発行して新たなデータを取得して、その後、当該データを処理するというものである。

#### 【0287】

その後、ロックライン予約喪失イベント・ハンドラを終了する（ステップ3150）。チャンネル書き込み（wrch）命令をSPU\_WriteEventMask [mask] を有するSPU書き込みイベント・マスク・チャンネルへ発行することによって、「マスク」を復元する（ステップ3160）。その後、一般イベント・ハンドラを終了して（ステップ3170）、処理が終わる。

#### 【0288】

##### 特権的アテンション・イベント

特権的アテンション・イベントは、特権的ソフトウェアがSPUプログラムからのアテンションを要求していることをSPUプログラムに通知するために使用される。特権的ソフトウェアは、SPU特権的制御レジスタにおけるアテンション・イベントを必要とするビットに「1」を書き込む（すなわち、SPU\_PrivCntl [A] が1に設定される）ことによって、アテンションを要求する。イベントがイネーブル（すなわち、SPU\_ReadEventStat [A] が「1」に設定されている）である場合であって、SPU読み出しイベント・ステータス・チャンネルについてのカウンタが「1」に設定されている場合に、このイベントは、Pend\_Event [A] ビットを「1」に設定する。チャンネル書き込み（wrch）が、Aビットが「1」に設定された（すなわち、SPU\_WriteEventAck [A] が「1」である）SPU書き込みイベント確認チャンネルに発行される場合、または、特権的ソフトウェアが、対応ビットが「0」に設定されたSPUチャンネル・アクセス・ファシリティを使用してSPU保留イベント・レジスタを更新する場合に、Pend\_Event [A] ビットは「0」に設定される。

#### 【0289】

特権的アテンション・イベントを処理する手順を図65に概略的に示す。図65に示すように、手順は、チャンネル読み出し（rdch）命令をSPU読み出しイベント・マスク・チャンネルへ発行して、データを「マスク」に退避することによって開始する（ステ

10

20

30

40

50

ップ3210)。イベントは、SPU\_WrEventMask[A]が「0」に設定されたSPU書き込みイベント・マスク・チャンネルに、チャンネル書き込み(wrch)命令を発行することによってマスクされる(ステップ3220)。イベントは、SPU\_WrEventAck[A]が「1」に設定されたSPU書き込みイベント確認チャンネルに対してチャンネル書き込み(wrch)命令を実行することによって確認される(ステップ3230)。

#### 【0290】

特権的アテンション・イベントに応じて、アプリケーション別の機能を実行する(ステップ3240)。これは、例えば、SPUの産物または何らかの他の行為が要求されているを知らせるために使用することができる。停止、知らせ、SPUインバウンド・メールボックス書き込み、SPUアウトバウンド割り込みメールボックス書き込み、またはシステムまたはI/Oメモリシステム内のステータスの更新などといった、特権的アテンション・イベントへのアプリケーションまたはオペレーティング・システム専用の応答が発行されなければならない。

#### 【0291】

特権的アテンション・イベント・ハンドラを終了する(ステップ3250)。チャンネル書き込み(wrch)命令をSPU\_WrEventMask[mask]を有するSPU書き込みイベント・マスク・チャンネルへ発行することによって、「マスク」を復元する(ステップ3260)。その後、一般イベント・ハンドラを終了する(ステップ3270)。

#### 【0292】

##### マルチソース同期イベント

マルチソース同期イベントは、マルチソース同期要求が完了した旨をSPUプログラムに通知するために使用される。マルチソース同期は、MFC書き込みマルチソース同期要求チャンネル(MFC\_WrMSSyncReq)への書き込み(wrch)によって要求される。このイベントは、MFC書き込みマルチソース同期要求チャンネル(MFC\_WrMSSyncReq)についてのチャンネル・カウントが「0」から「1」の値へ変化すると生じる。このイベントがイネーブル(すなわち、SPU\_RdEventStat[Ms]が「1」)である場合であって、SPU読み出しイベント・ステータス・チャンネルについてのカウントが「1」に設定されている場合に、このイベントは、Pend\_Event[Ms]ビットを「1」に設定する。チャンネル書き込み(wrch)が、Msビットが設定された(すなわち、SPU\_WrEventAck[Ms]が「1」に設定された)SPU書き込みイベント確認チャンネルに発行される場合、または、特権的ソフトウェアが、対応ビットが「0」に設定されたSPUチャンネル・アクセス・ファシリティを使用してSPU保留イベント・レジスタを更新する場合に、Pend\_Event[Ms]ビットは「0」に設定される。マルチソース同期イベントは、マルチソース同期要求が発行されるメッセージ前にクリアされなくてはならない。

#### 【0293】

マルチソース同期イベントを処理する手順を図66に概略的に示す。図66に示すように、手順は、チャンネル読み出し(rdch)命令をSPU読み出しイベント・マスク・チャンネルへ送って、データを「マスク」に退避することによって開始する(ステップ3310)。イベントは、SPU\_WrEventMask[Tm]が「0」に設定されたSPU書き込みイベント・マスク・チャンネルに、チャンネル書き込み命令を発行することによってマスクされる(ステップ3320)。イベントは、SPU\_WrEventAck[Ms]が「1」に設定されたSPU書き込みイベント確認チャンネルに対してチャンネル書き込み(wrch)命令を実行することによって確認される(ステップ3330)。

#### 【0294】

保留マルチソース同期動作の完了に応じて、アプリケーション別の機能を実行する(ステップ3340)。これは、典型的には、例えば、特定のバッファ内のデータが完全に更

10

20

30

40

50

新されたこと、またはバッファ領域はもはや使用されていないことを示すことである。マルチソース同期イベント・ハンドラを終了する（ステップ3350）。チャンネル書き込み（w r c h）命令をS P U \_ W r E v e n t M a s k [ m a s k ] を有するS P U書き込みイベント・マスク・チャンネルへ発行することによって、「マスク」を復元する（ステップ3360）。その後、一般イベント・ハンドラを終了する（ステップ3370）。

#### 【0295】

要約すると、本発明は、プロセッサと、ローカル記憶の負担を軽減しかつプロセッサがデータ、空間の開放、またはイベントの発生を待つときに低電力状態でいられる外部装置との間の通信を促進するための機構を提供する。本発明の機構は、プロセッサ、メモリ・フロー・コントローラ、マシン・ステート・レジスタ、および外部プロセッサ割り込みファシリティなどの様々なファシリティとの通信のための複数のチャンネルを提供する。これらのチャンネルは、命令、命令パラメータ、内部プロセッサ情報、信号通知、マシン分離情報、マシン割り込み状態情報、生成されたイベントを提供し、またイベント処理を行うのに使用されてもよい。

#### 【0296】

本発明を、完全に機能するデータ処理システムにおいて説明してきたが、本発明の処理は、コンピュータ読み取り可能な媒体の命令の形式および様々な形式で分散できること、また、本発明は、分散を実行するのに実際に使用される、信号が記録された媒体の特定の種類に関係なく等しく当てはまることを、当業者は理解するだろうということに留意することが重要である。コンピュータ読み取り可能な媒体の例としては、フレキシブルディスク、ハードディスクドライブ、RAM、CD-ROM、DVD-ROMなどといった記録可能型の媒体、ならびに、例えば無線周波数および光波送信などの送信形式を使用するデジタルおよびアナログ通信リンクおよび有線または無線の通信リンクなどといった送信型の媒体が含まれる。コンピュータ読み取り可能な媒体は、符号化された形式を取ってもよく、この復号化された形式は、特定のデータ処理システムにおける実際の使用のためには復号化される。

#### 【0297】

本発明の説明を例示および説明の目的で提示してきたが、網羅的なものでもなく、開示された形式に本発明を限定するものでもない。数多くの修正および変更が当業者にとって明らかであろう。本発明の原理および実際の応用を最もよく説明するため、また、予期された特定の使用に適した様々な変更を伴う様々な実施形態について当業者が本発明を理解することができるようにするために、選択かつ説明されたものである。

#### 【図面の簡単な説明】

#### 【0298】

【図1】本発明の例示的な一実施形態に係るブロードバンド・プロセッサ・アーキテクチャ（BPA）のブロック図の例である。

【図2】本発明の例示的な一実施形態に係る典型的なMFC200のブロック図の例である。

【図3】本発明の例示的な一実施形態に係るMFCコマンドについてのニーモニック・パラメータを示す図の例である。

【図4】本発明の例示的な一実施形態に係るDMAリスト・コマンドの詳細を示す図の例である。

【図5】米国特許出願公開第2004/0264445号に記載された機構に係るチャンネルの単一の対のためのチャンネル回路に関するSPU発行/制御論理の仕組みならびにデータ・フローを示す図の例である。

【図6】本発明の例示的な一実施形態に係るチャンネル・インターフェースの動作の例の概略を示すフローチャートである。

【図7】チャンネルが本発明の一実施形態によって使用されるやり方を示す図の例である。

10

20

30

40

50

【図 8】本発明の例示的な一実施形態に係る S P U チャンネル・マップを表で示す図の例である。

【図 9】本発明の例示的な一実施形態に係る S P U チャンネル・マップを表で示す図の例である。

【図 10】本発明の例示的な一実施形態に係る S P U チャンネル・マップを表で示す図の例である。

【図 11】本発明の例示的な一実施形態に係る S P U チャンネル・マップを表で示す図の例である。

【図 12】本発明の例示的な一実施形態に係る M F C コマンド操作符号チャンネルの詳細を示す図の例である。

10

【図 13】本発明の例示的な一実施形態に係る M F C クラス I D チャンネルの詳細を示す図の例である。

【図 14】本発明の例示的な一実施形態に係る 8 ウェイ・セット・アソシエティブ・キャッシュに対する典型的な R M T エントリを示す。

【図 15】本発明の例示的な一実施形態に係る M F C コマンド・タグ識別チャンネルの詳細を示す図の例である。

【図 16】本発明の例示的な一実施形態に係る M F C 転送サイズまたはリスト・サイズ・チャンネルの詳細を示す図の例である。

【図 17】本発明の例示的な一実施形態に係る M F C ローカル記憶アドレス・チャンネルの詳細を示す図の例である。

20

【図 18】本発明の例示的な一実施形態に係る M F C 有効アドレス l o w またはリスト・アドレス・チャンネルの詳細を示す図の例である。

【図 19】本発明の例示的な一実施形態に係る M F C 有効アドレス h i g h チャンネルの詳細を示す図の例である。

【図 20】本発明の例示的な一実施形態に係る M F C コマンド・パラメータを書き込むための動作の例の概略を示す図の例である。

【図 21】本発明の例示的な一実施形態に係る M F C 書き込みタグ・グループ・クエリ・マスク・チャンネルの詳細を示す図の例である。

【図 22】本発明の例示的な一実施形態に係る M F C 読み出しタグ・グループ・クエリ・マスク・チャンネルの詳細を示す図の例である。

30

【図 23】本発明の例示的な一実施形態に係る M F C 書き込みタグ・ステータス更新要求チャンネルの詳細を示す図の例である。

【図 24】本発明の例示的な一実施形態に係る M F C 読み出しタグ・ステータス更新要求チャンネルの詳細を示す図の例である。

【図 25】本発明の例示的な一実施形態に係る M F C 読み出しリスト・ストール／通知タグ・ステータスチャンネルの詳細を示す図の例である。

【図 26】本発明の例示的な一実施形態に係る M F C 書き込みリスト・ストール／通知タグ・ステータスチャンネルの詳細を示す図の例である。

【図 27】本発明の例示的な一実施形態に係る M F C 読み出しアトミック・コマンド・ステータス・チャンネルの詳細を示す図の例である。

40

【図 28】本発明の例示的な一実施形態に係る M F C コマンドの完了または M F C コマンド・グループの完了のためのポーリングのための動作の例の概略を示すフローチャートの例である。

【図 29】本発明の例示的な一実施形態に係るタグ・グループ更新待機またはイベント待機（1 つ以上のタグ・グループの完了）のための動作の例の概略を示すフローチャートの例である。

【図 30】本発明の例示的な一実施形態に係る条件付タグ・イベントの待機またはポーリングに代わるものとしての、S P U イベント・ファシリティを使用するための動作の例の概略を示すフローチャートの例である。

【図 31】本発明の例示的な一実施形態に係る、M F C ・ D M A リスト・コマンドがスト

50

ール／通知フラグ・セットを有するリスト要素に到達したかどうかを判断するためのポーリングのための動作の例の概略を示す図の例である。

【図 3 2】本発明の例示的な一実施形態に係る、M F C ・ D M A リスト・コマンドがストール／通知フラグ・セットを有するリスト要素に到達するのを待つための動作の例の概略を示す図の例である。

【図 3 3】本発明の例示的な一実施形態に係る、ストール／通知タグ・グループ・ステータスのリストを待機またはポーリングする代わりとして、S P U イベント・ファシリティを使用するための動作の例の概略を示す図の例である。

【図 3 4】本発明の例示的な一実施形態に係る M F C 書き込みマルチソース同期要求チャンネルの詳細を示す図の例である。

10

【図 3 5】本発明の例示的な一実施形態に係る S P U 書き込みアウトバウンド・メールボックス・チャンネルの詳細を示す図の例である。

【図 3 6】本発明の例示的な一実施形態に係る S P U 書き込みアウトバウンド割り込みメールボックス・チャンネルの詳細を示す図の例である。

【図 3 7】本発明の例示的な一実施形態に係る S P U 読み出しインバウンド・メールボックス・チャンネルの詳細を示す図の例である。

【図 3 8】本発明の例示的な一実施形態に係る S P U 信号通知チャンネルの詳細を示す図の例である。

【図 3 9】本発明の例示的な一実施形態に係る S P U 書き込みデクリメンタ・チャンネルの詳細を示す図の例である。

20

【図 4 0】本発明の例示的な一実施形態に係る S P U 読み出しデクリメンタ・チャンネルの詳細を示す図の例である。

【図 4 1】本発明の例示的な一実施形態に係る S P U 読み出しマシン・ステータス・チャンネルの詳細を示す図の例である。

【図 4 2】本発明の例示的な一実施形態に係る S P U 書き込み状態退避／復元チャンネルの詳細を示す図の例である。

【図 4 3】本発明の例示的な一実施形態に係る S P U 読み出し状態退避／復元チャンネルの詳細を示す図の例である。

【図 4 4】本発明の例示的な一実施形態に係る S P U イベント・ファシリティの論理表現を示すブロック図の例である。

30

【図 4 5】本発明の例示的な一実施形態に係る S P U イベント・ファシリティの論理表現を示すブロック図の例である。

【図 4 6】本発明の例示的な一実施形態に係る S P U 読み出しイベント・ステータス・チャンネルの詳細を示す図の例である。

【図 4 7】本発明の例示的な一実施形態に係る S P U 書き込みイベント・マスク・チャンネルの詳細を示す図の例である。

【図 4 8】本発明の例示的な一実施形態に係る S P U 読み出しイベント・マスク・チャンネルの詳細を示す図の例である。

【図 4 9】本発明の例示的な一実施形態に係る S P U 書き込みイベント確認チャンネルの詳細を示す図の例である。

40

【図 5 0】本発明の例示的な一実施形態に係る S P U 書き込みイベント確認チャンネルの詳細を示す図の例である。

【図 5 1】本発明の例示的な一実施形態に係る S P U 書き込みイベント確認チャンネルの詳細を示す図の例である。

【図 5 2】本発明の例示的な一実施形態に係る S P U 書き込みイベント確認チャンネルの詳細を示す図の例である。

【図 5 3】本発明の例示的な一実施形態に係る S P U 書き込みイベント確認チャンネルの詳細を示す図の例である。

【図 5 4】本発明の例示的な一実施形態に係る S P U 書き込みイベント確認チャンネルの詳細を示す図の例である。

50

【図 5 5】本発明の例示的な一実施形態に係る S P U 書き込みイベント確認チャンネルの詳細を示す図の例である。

【図 5 6】本発明の例示的な一実施形態に係る M F C ・ D M A リスト・コマンド・ストール／通知イベントを処理するための動作の例の概略を示すフローチャートである。

【図 5 7】本発明の例示的な一実施形態に係る M F C ・ S P U コマンド・キュー使用可能イベントを処理するための動作の例の概略を示すフローチャートである。

【図 5 8】本発明の例示的な一実施形態に係る S P U インバウンドメールボックス使用可能イベントを処理するための動作の例の概略を示すフローチャートである。

【図 5 9】本発明の例示的な一実施形態に係る S P U デクリメンタ・イベントを処理するための動作の例の概略を示すフローチャートである。

10

【図 6 0】本発明の例示的な一実施形態に係る S P U アウトバウンド割り込みメールボックス使用可能イベントを処理するための動作の例の概略を示すフローチャートである。

【図 6 1】本発明の例示的な一実施形態に係る S P U アウトバウンド・メールボックス使用可能イベントを処理するための動作の例の概略を示すフローチャートである。

【図 6 2】本発明の例示的な一実施形態に係る S P U 信号通知 2 使用可能イベントを処理するための動作の例の概略を示すフローチャートである。

【図 6 3】本発明の例示的な一実施形態に係る S P U 信号通知 1 使用可能イベントを処理するための動作の例の概略を示すフローチャートである。

【図 6 4】本発明の例示的な一実施形態に係るロック・ライン予約喪失イベントを処理するための動作の例の概略を示すフローチャートである。

20

【図 6 5】本発明の例示的な一実施形態に係る特権的アテンション・イベントを処理するための動作の例の概略を示すフローチャートである。

【図 6 6】本発明の例示的な一実施形態に係るマルチソース同期イベントを処理するための動作の例の概略を示すフローチャートである。

【符号の説明】

【 0 2 9 9 】

1 2 0 , 1 2 2 , 1 2 4 , 1 2 6 , 1 2 8 , 1 3 0 , 1 3 2 , 1 3 4    S P E

1 4 0 , 1 4 2 , 1 4 4 , 1 4 6 , 1 4 8 , 1 5 0 , 1 5 2 , 1 5 4    S P U

1 6 3 ~ 1 6 7    L S

1 5 5 ~ 1 6 2    M F C

30

1 8 0 , 1 8 2 , 1 8 4 , 1 8 6 , 1 8 8 , 1 9 0 , 1 9 2 , 1 9 4    B I U

1 1 2    L 2 キャッシュ

1 1 4    L 1 キャッシュ

1 1 6    P P U

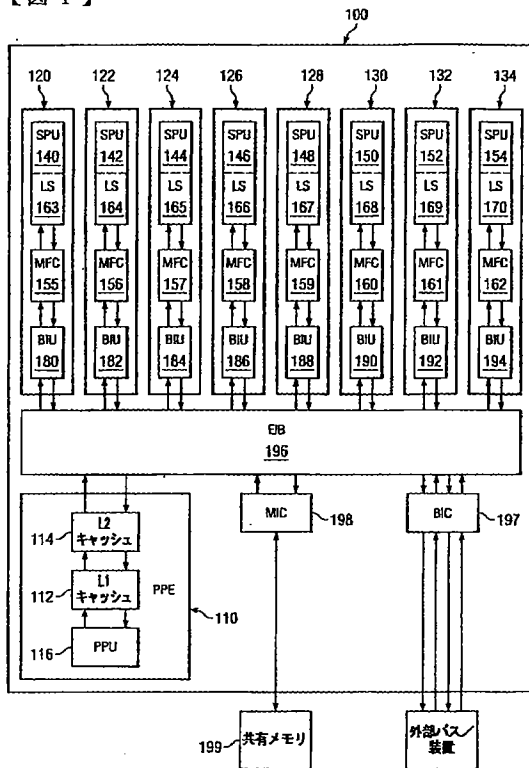
1 9 6    E I B

1 9 7    B I U

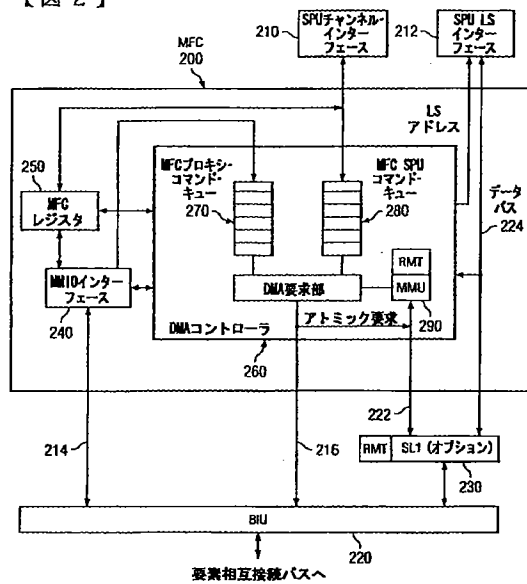
1 9 8    M I C

1 9 9    共有メモリ

【図 1】



【図 2】



【図 3】

パラメータ	パラメータ名	レジスタ名	注記 参照
CL	MFCクラスID	MFC_ClassID	
TG	MFCコマンドタグ識別	MFC_TagID	
TS	MFC転送サイズ	MFC_Size	1
LSZ	MFCリストサイズ	MFC_Size	1
LSA	MFCローカル記憶アドレス	MFC_LSA	
EAH	MFC有効アドレスHIGH	MFC_EAH	4
EAL	MFC有効アドレスLOW	MFC_EAL	2
LA	MFCリスト・ローカル記憶アドレス	MFC_EAL	2
LTS	リスト要素転送サイズ		3
LEAL	リスト要素有効アドレスLOW		3

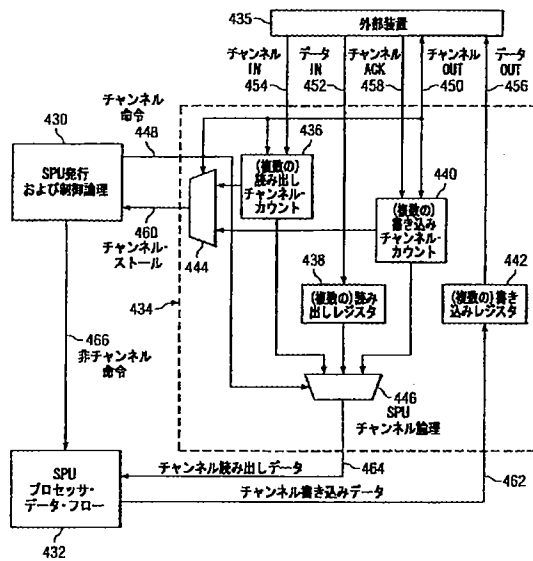
1. TSおよびLSZは、同一のレジスタ・オフセットを共有  
 2. EALおよびLAは、同一のレジスタ・オフセットを共有  
 3. 関連レジスタはなし。これらのパラメータは、ローカル記憶にあり、リスト・アドレス(LA)・パラメータによって参照される  
 4. このパラメータはオプションである

【図 4】

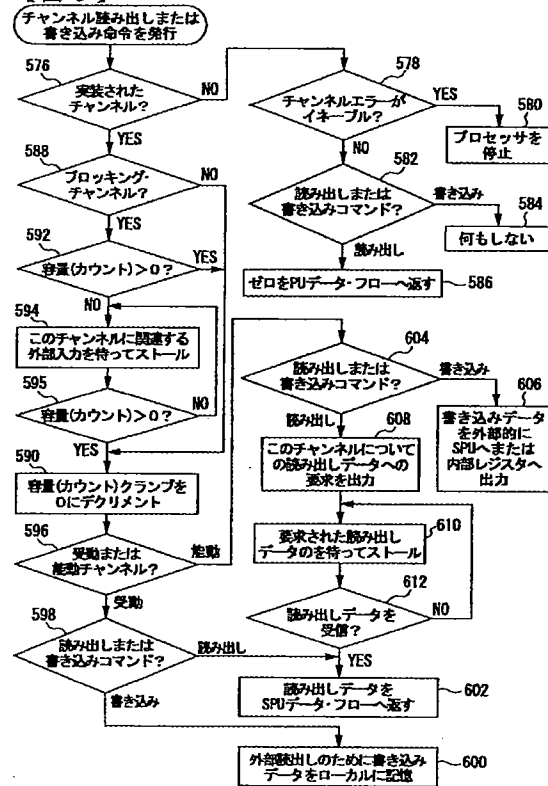
ビット	フィールド名	説明
32	0	ストローブル通知ビット
31	1:16	予約
30	17:31	リスト要素転送サイズ (LTS)
29	32:63	64ビット有効アドレスのLOWワード (LEAL)



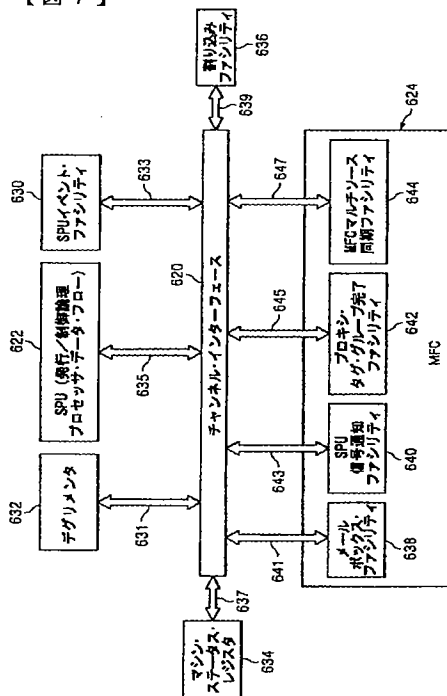
【図 5】



【図 6】



【図 7】



【図 8】

チャンネル番号 (16進数)	チャンネル名	説明	アクセス種類
X0'	SPU_ReadEventStat	SPU読み出しイベントステータスを読み出し (マスタクを適用)	読み出し
X1'	SPU_WriteEventMask	SPU書き込みイベントステータスマスクを書き込み	書き込み
X2'	SPU_WriteEventAck	SPU書き込みイベントステータスマスクをクリア	書き込み
X3'	SPU_ReadSignal1	SPU信号通知1チャンネル	読み出し
X4'	SPU_ReadSignal2	SPU信号通知2チャンネル	読み出し
X5'	チャンネル5	予備	読み出し
X6'	チャンネル6	予備	読み出し
X7'	SPU_ReadSignal3	SPU信号通知3チャンネル	書き込み
X8'	SPU_ReadSignal4	SPU信号通知4チャンネル	書き込み
X9'	SPU_ReadSignal5	SPU信号通知5チャンネル	読み出し
X0''	MFC_WriteSyncReq	MFC書き込み要求チャンネル	書き込み
X1''	MFC_ReadSyncReq	MFC読み出し要求チャンネル	読み出し
X2''	MFC_WriteSyncReq	MFC書き込み要求チャンネル	書き込み
X3''	MFC_ReadSyncReq	MFC読み出し要求チャンネル	読み出し

【図 9】

SPU予備チャンネル			
XA	チャンネルID	予備	
マスク読み込みチャンネル			
X8	SPU_RdEmulMask	SPU読み出しイベント・マスク・チャンネル	読み出し
X9	MFC_RdTagMask	MFC読み出しタグ・グループ・マスク・チャンネル	読み出し
SPU状態管理チャンネル			
XD	SPU_RdMachStat	SPU読み出しマシン・ステータス・チャンネル	読み出し
XE	SPU_WrSRR0	SPU書き込み状態管理/復元チャンネル	書き込み
XF	SPU_RdSRR0	SPU読み出し状態管理/復元チャンネル	読み出し
MFCコマンド・パラメータ・チャンネル			
X10	MFC_LSA	MFCローカル距離アドレス・チャンネル ローカル距離アドレス・コマンド・パラメータを書き込み	書き込み
X11	MFC_EAH	MFC有効アドレスH(8)チャンネル 高8ビットのMFC有効アドレス・コマンド・パラメータを書き込み	書き込み
X12	MFC_EAL	MFC有効アドレスL(8)チャンネル 低8ビットのMFC有効アドレス・コマンド・パラメータを書き込み	書き込み
X13	MFC_Size	MFC転送サイズまたはリスト・サイズ・チャンネル MFC転送サイズ・コマンド・パラメータを書き込み	書き込み

図 8 から

【図 11】

SPUメールボックス			
チャンネル	SPU書き込み/アウトバウンド・メールボックス・チャンネル	予備	
X1C	SPU_WrOutMbox	SPU書き込み/アウトバウンド・メールボックス内容を書き込み	WRITE-BLOCKING
X1D	SPU_RdInMbox	SPU読み出し/インバウンド・メールボックス・チャンネル	READ-BLOCKING
X1E	SPU_RdOutMbox	SPU読み出し/アウトバウンド・メールボックス内容を読み出し	WRITE-BLOCKING
X1F ~ X3F	SPU_WrInMbox	SPU書き込み/インバウンド・メールボックス内容を書き込み	WRITE-BLOCKING
チャンネル63	SPU_WrOutMbox	SPU書き込み/アウトバウンド・メールボックス内容を書き込み	WRITE-BLOCKING

図 10 から

668

【図 10】

チャンネル番号 (18ビット)	チャンネル名	説明	アクセス種類
X14	MFC_TagID	MFCコマンドタグ識別チャンネル タグ識別子・コマンド・パラメータを書き込み	書き込み
X15	MFC_Cmd MFC_ClassID	MFCコマンド操作符チャンネル MFCコマンドを識別クラスIDに書き込みおよびエンキュー MFCクラスIDチャンネル MFCコマンドを識別コマンド操作符に書き込みおよびエンキュー	書き込み ブロッキング
MFCタグ・ステータス・チャンネル			
X16	MFC_WrTagMask	MFC書き込みタグ・グループ・マスク・チャンネル タグ・マスクを書き込み	書き込み
X17	MFC_WrTagUpdate	MFC書き込みタグ・ステータス更新要求チャンネル 条件または無条件のタグ・ステータス更新の要求を書き込み	書き込み ブロッキング
X18	MFC_RdTagStat	MFC読み出しタグ・グループ・ステータス・チャンネル タグ・ステータスを読み出し (マスクを適用)	読み出し ブロッキング
X19	MFC_RdInStatStat	MFC読み出し/インバウンド・ステータス・チャンネル MFC読み出し/インバウンド・ステータスを読み出し	読み出し ブロッキング
X1A	MFC_WrInStatStat	MFC書き込み/インバウンド・ステータス・チャンネル MFC書き込み/インバウンド・ステータスを書き込み	書き込み ブロッキング
X1B	MFC_RdOutStat	MFC読み出し/アウトバウンド・ステータス・チャンネル MFC読み出し/アウトバウンド・ステータスを読み出し	読み出し ブロッキング

図 11 から

【図 12】

アクセス種類		書き込みブロッキング	
チャンネル番号		X'15' (下位16ビット)	
予備		MFCコマンド操作符	
ビット		フィールド名	
0:7		予備	
8:15		MFC コマンド 操作符	
		説明	
		通常はゼロに設定されなければならない 「1」に設定されたビットゼロは、 操作符が予約されたことを示す	
		MFCコマンド操作符	

【図 13】

アクセス種類		書き込みブロッキング	
チャンネル番号		X'15' (上位16ビット)	
TCLASSID		RCLASSID	
ビット		フィールド名	
0:7		TCLASSID	
8:15		RCLASSID	
		説明	
		転送クラス識別子	
		置換クラス識別子	

【図 14】

アクセス種類	書き込み	チャンネル番号	X'10'
ビット	フィールド名	説明	
0:1	予約	ゼロに設定	
2:3	予約	ゼロに設定	
4:5	予約	ゼロに設定	
6:7	予約	ゼロに設定	
8:9	予約	ゼロに設定	
10:11	予約	ゼロに設定	
12:13	予約	ゼロに設定	
14:15	予約	ゼロに設定	
16:17	予約	ゼロに設定	
18:19	予約	ゼロに設定	
20:21	予約	ゼロに設定	
22:23	予約	ゼロに設定	
24:25	予約	ゼロに設定	
26:27	予約	ゼロに設定	
28:29	予約	ゼロに設定	
30:31	予約	ゼロに設定	

【図 15】

アクセス種類	書き込み	チャンネル番号	X'14'
ビット	フィールド名	説明	
0:10	予約	ゼロに設定	
11:15	MFCコマンド・タグ識別	MFCコマンド・タグ識別	

【図 16】

アクセス種類	書き込み	チャンネル番号	X'13'
ビット	フィールド名	説明	
0	予約	ゼロに設定	
1:15	MFC転送・リストサイズ	MFC転送サイズまたはリストサイズ。 有効な範囲は： ・ 0 ≤ MFC転送サイズ ≤ リストサイズが16KBまたは2KB ・ 0 ≤ MFC転送サイズ ≤ リストサイズが16KBまたは2KB 許容MFC転送／リストサイズは： ・ (自然数)1, 2, 4, および8バイトであって、 送元アドレスおよび送先アドレスは、同一の 最小4ビットを有しなければならない。 (DMAアラインメントについての異なる情報は、 PowerPCアーキテクチャのブックIIを参照) ・ 16バイトおよび16バイトから16KBまでの倍数であって、 送元アドレスおよび送先アドレスは、整列した 16バイト(クワッドワード)でなければならない。 許容DMAリストサイズは： ・ リストサイズについては、最小サイズが8バイトで あって、最大サイズが16KBの8バイトの倍数で あって、リストは、ローカル記憶において8バイト (ダブルワード)の境界で開始しなければならない。 これにより、1個から2048個までの要素を取り得る リストが提供される	

【図 17】

アクセス種類	書き込み	チャンネル番号	X'10'
ビット	フィールド名	説明	
0:31	MFC ローカル記憶 アドレス	MFCローカル記憶アドレス。 注：ローカル記憶アドレスの最小桁4ビットと 一致しなければならない。	

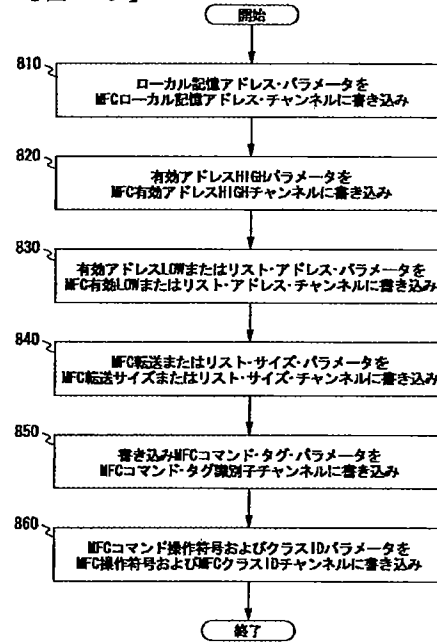
【図 18】

アクセス種類	書き込み	チャンネル番号	X'12'
ビット	フィールド名	説明	
0:31	MFC ローカル記憶 アドレス	64ビット有効アドレスまたはMFCリストの ローカル記憶アドレスのLOWワード。 注：ローカル記憶アドレスの最小桁4ビットと 一致しなければならない。	

【図 19】

アクセス種類 チャンネル番号		書き込み X'11'		64ビット有効アドレスの HIGHワード (オプション)																											
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ビット		フィールド名		説明																											
0-31		64ビット有効アドレスのHIGHワード		64ビット有効アドレスのHIGHワードは、有効アドレスの上位32ビットに等しい。有効アドレスの低位32ビットは、有効アドレスの低位32ビットに等しい。有効アドレスの高位32ビットは、有効アドレスの高位32ビットに等しい。有効アドレスの低位32ビットは、有効アドレスの低位32ビットに等しい。																											

【図 20】



【図 21】

アクセス種類 チャンネル番号		書き込み X'15'		タグ・グループ「n」選択																															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
ビット		フィールド名		説明																															
0:31		q <sub>n</sub>		タグ・グループ「n」選択 0 タグ・グループは、クエリまたはタグ・イベントの一部ではない。 1 タグ・グループは、クエリまたはタグ・イベントの一部である。																															

【図 22】

アクセス種類 チャンネル番号		読み出し X'6'																																	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
ビット		フィールド名		説明																															
0:31		0 <sub>n</sub>		タグ・グループ「n」選択 0 タグ・グループは、クエリまたはタグ・イベント・タグ・グループの一部ではない 1 タグ・グループは、クエリまたはタグ・イベント・タグ・グループの一部である																															

【図 23】

アクセス権限		書き込みプロテクト	
チャンネル番号		X'17	
ビット		説明	
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31		TS	
ビット		フィールド名	
0:29		予約	
30:31		TS	
		説明	
		タグ・ステータス更新条件	
		00 すぐに更新 無条件	
		01 いずれかのイネーブルなタグ・グループが「未処理動作なし」ステータスの場合、	
		タグ・ステータスを更新	
		10 すべてのイネーブルなタグ・グループが「未処理動作なし」ステータスの場合、	
		タグ・ステータスを更新	
		11 予約	

【図 24】

アクセス権限		読み出しプロテクト	
チャンネル番号		X'18	
ビット		説明	
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31		タグ・グループ「n」ステータス	
		1 タグ・グループは、未処理の動作を有しておらず（コマンドは完了）、	
		クエリ・マスクによってステータスエネーブルになっているか	
		0 タグ・グループは、未処理の動作を有しているが、クエリ・マスクによって	
		ステータスエネーブルになっている	

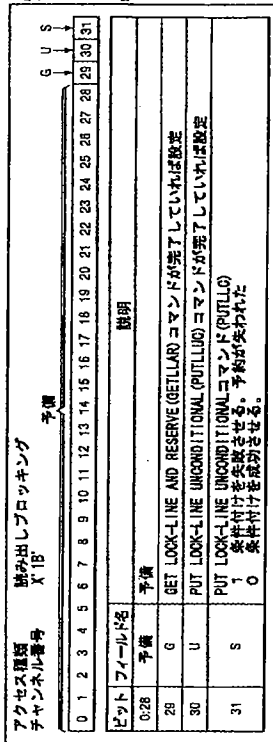
【図 25】

アクセス権限		読み出しプロテクト	
チャンネル番号		X'19	
ビット		説明	
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31		タグ・グループ「n」ステータス	
		1 タグ・グループは、ステータス・通知フラグが設定された状態でステータスとして	
		いる非リセットコマンドを有する	
		0 タグ・グループは、現在ステータスとして非リセットコマンドを有しない	

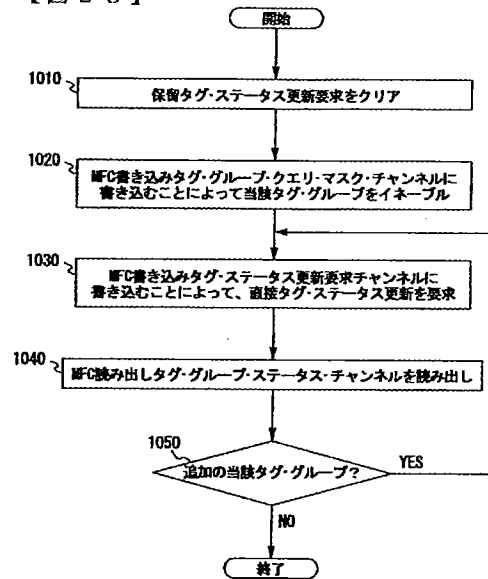
【図 26】

アクセス権限		書き込み	
チャンネル番号		X'1A	
ビット		説明	
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31		タグ・グループ「n」ステータス	
		1 タグ・グループは、ステータス・通知フラグが設定された状態でステータスとして	
		いる非リセットコマンドを有する	
		0 タグ・グループは、現在ステータスとして非リセットコマンドを有しない	

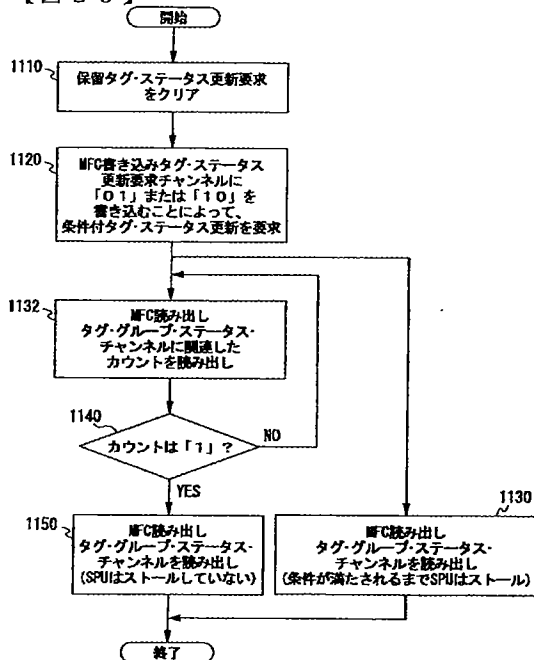
【図 27】



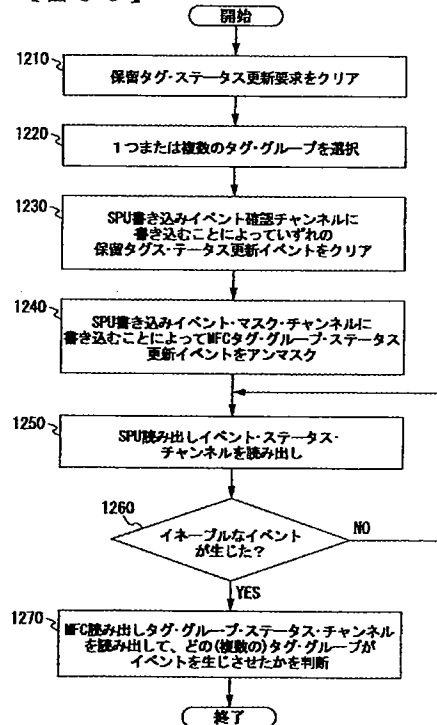
【図 28】



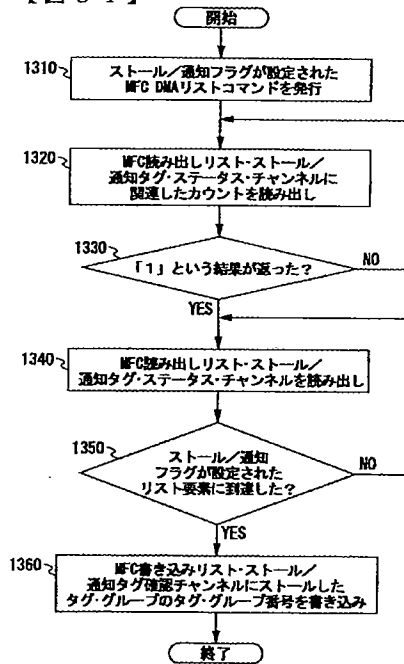
【図 29】



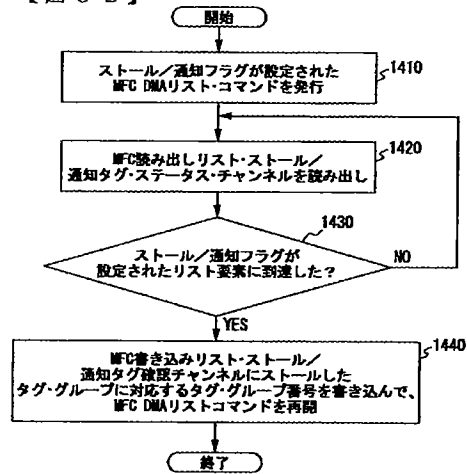
【図 30】



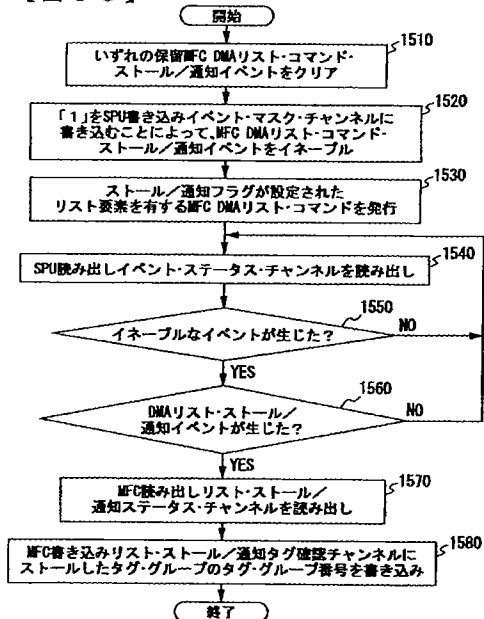
【図 3 1】



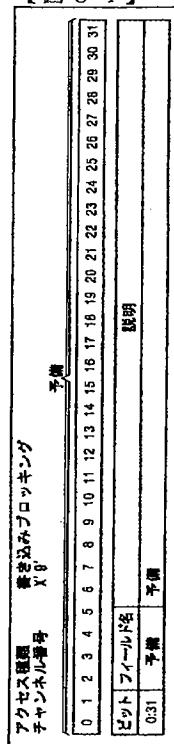
【図 3 2】



【図 3 3】



【図 3 4】









【図 4 3】

アクセス種類		読み出し		SPUデータ	
チャンネル番号		X/F		説明	
ビットフィールド名	ビット	状態遷移/表示レジスタ0データ		説明	
0.31	0	SPUデータ		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	

【図 4 4】

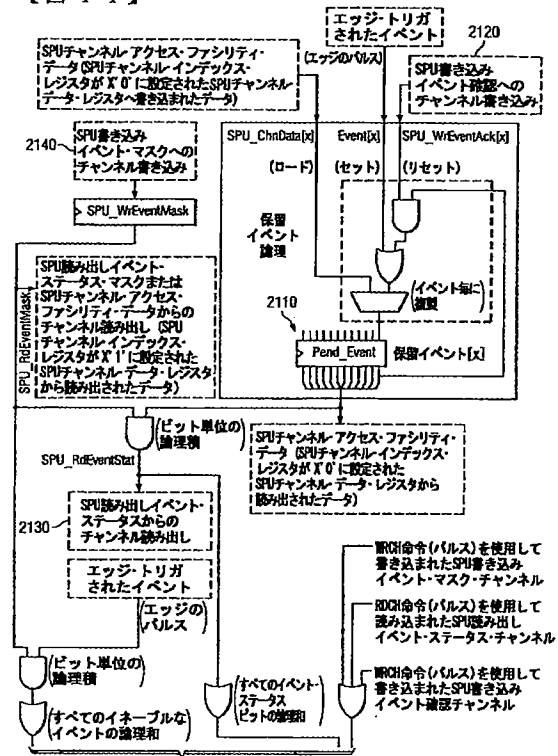
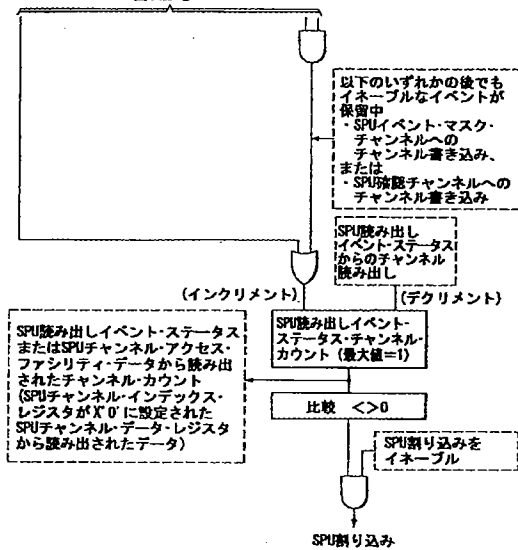


図45へ

【図 4 5】

図44から



【図 4 6】

アクセス種類		読み出し		SPUデータ	
チャンネル番号		X/F		説明	
ビットフィールド名	ビット	状態遷移/表示レジスタ0データ		説明	
0.18	0	SPUデータ		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	
19	19	SPUデータ		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	
20	20	SPUデータ		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	

図47へ



【图 5 1】

24	L8	SFU 0 イベントはデイスユーブル 1 イベントはイネーブル
25	M6	SFUアウトバウンド、メモリ込みメールボックス使用可能イベント・イネーブル 0 イベントはデイスユーブル 1 イベントはイネーブル
26	Tm	SFUデクリメンタにエラー発生した場合、デクリメント・イベント・カウンタをリセットする前に、デクリメント・イベント・カウンタはゼロになる。 0 イベントはデイスユーブル 1 イベントはイネーブル
27	Mb	SFUインバウンド、メールボックス使用可能イベント・イネーブルであって、PPEに よって書き込まれる 0 イベントはデイスユーブル 1 イベントはイネーブル
28	Ov	NFG SPULコマンドは、キュー使用可能イベント・イネーブル 0 イベントはデイスユーブル 1 イベントはイネーブル
29	予備	予備
30	Sn	NFG DMAリスト・コマンド・ストローノ通知イベント・イネーブル 0 イベントはデイスユーブル 1 イベントはイネーブル
31	Taq	NFGタグ・グループ・ステータス更新イベント・イネーブル 0 イベントはデイスユーブル 1 イベントはイネーブル

【图 5 2】

[illegible]

【图 5 3】

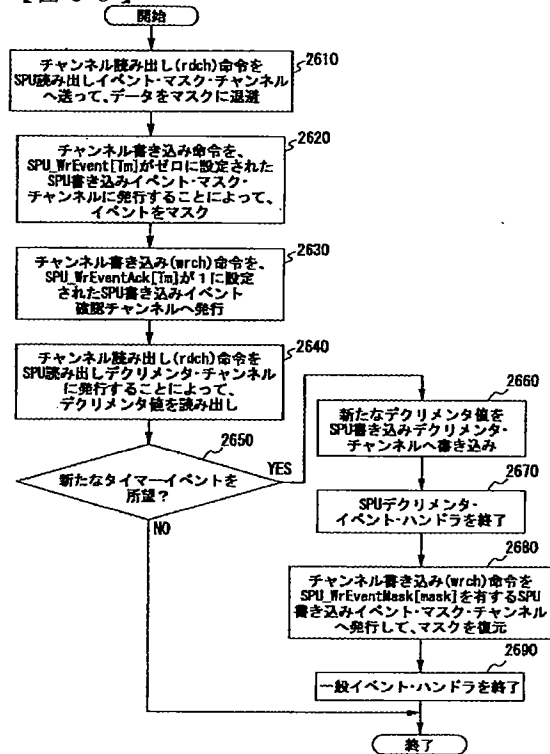
24	Le	SPUアウトバウンド: メールボックス使用可能イベント・イネーブル 0 イベントはデイスエーブル 1 イベントはイネーブル
25	Me	SPUアウトバウンド: 取り込みメールボックス使用可能イベント・イネーブル 0 イベントはデイスエーブル 1 イベントはイネーブル
26	Tm	SPUデクリメンタ: イベント・イネーブル 0 イベントはデイスエーブル 1 イベントはイネーブル
27	Mb	SPUインバウンド: メールボックス使用可能イベント・イネーブルであって、PREIN よって書き込まれる 0 イベントはデイスエーブル 1 イベントはイネーブル
28	Qv	MFC SPUコマンド: キュー使用可能イベント・イネーブル 0 イベントはデイスエーブル 1 イベントはイネーブル
29	予備	予備
30	Sh	MFC DMAリス: トコマンド: ストール/通知イベント・イネーブル 0 イベントはデイスエーブル 1 イベントはイネーブル
31	Taq	MFCタグ: グリッパステータス更新イベント・イネーブル 0 イベントはデイスエーブル 1 イベントはイネーブル

【图 5-4】

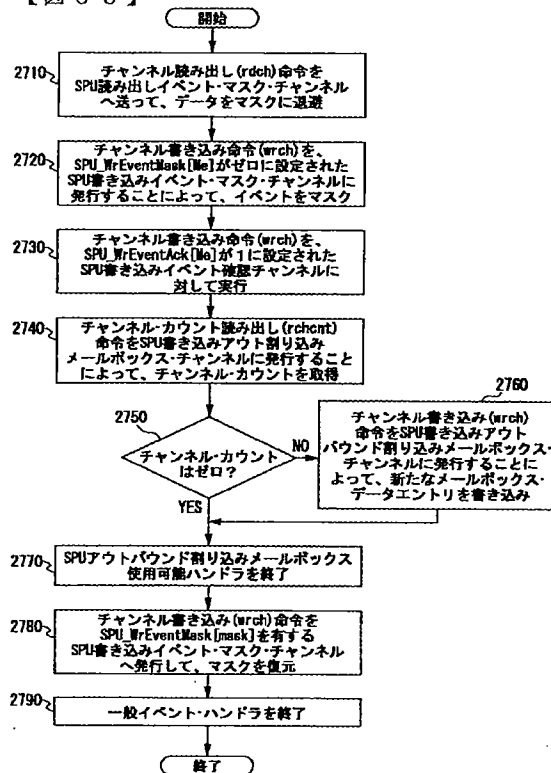
アクセル履歴 チャネル番号		書き込み X 2																予備															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
ビット フィールド名		予備																予備															
0:16		予備																説明 これらのイベントの詳細情報															
19	Ms	マルチソース同期イベント確認 0 イベントは未確認 1 イベントは確認																															
20	A	種横的アテンションイベント確認 0 イベントは未確認 1 イベントは確認																															
21	Lr	ロック/ライン丢失イベント確認 0 イベントは未確認 1 イベントは確認																															
22	S1	SP1番号通知1使用可能イベント確認 0 イベントは未確認 1 イベントは確認																															
23	S2	SP1番号通知2使用可能イベント確認 0 イベントは未確認 1 イベントは確認																															



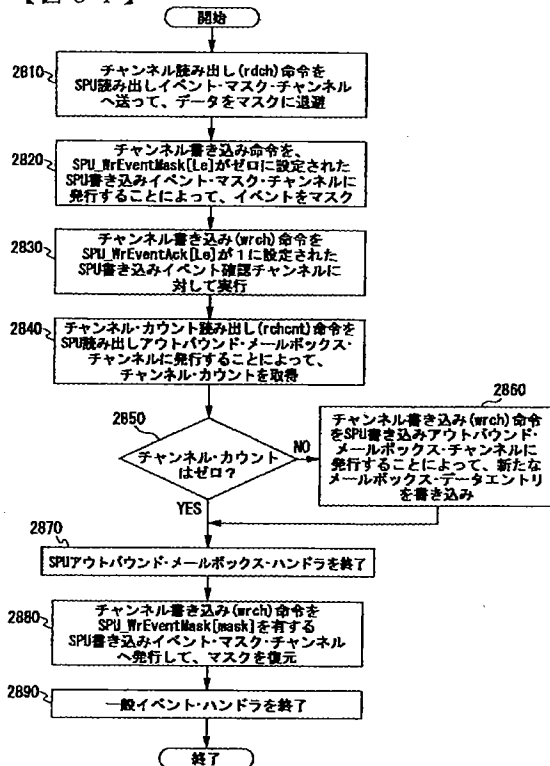
【図 59】



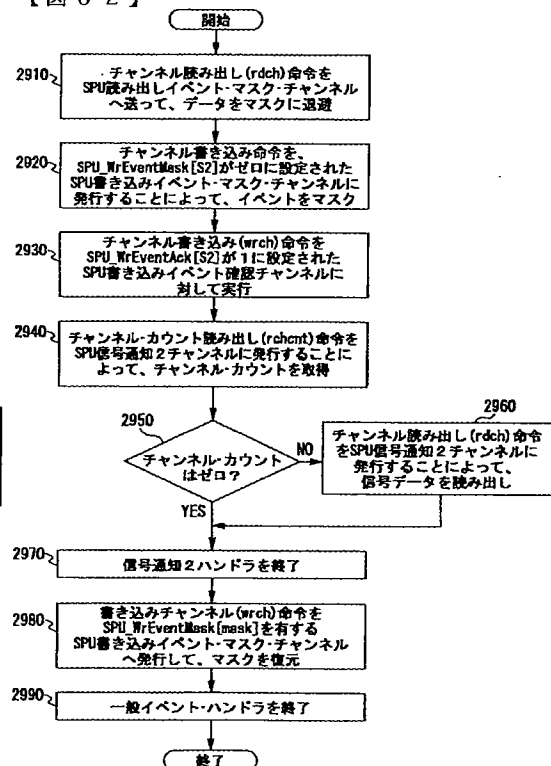
【図 60】



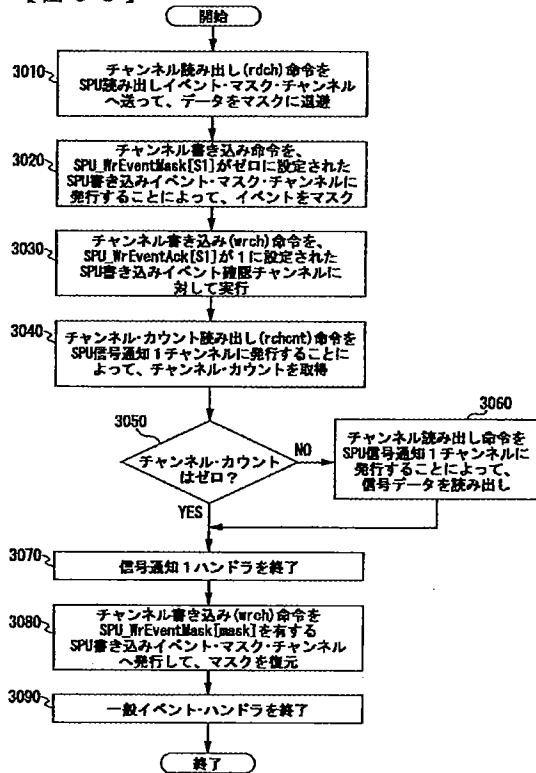
【図 61】



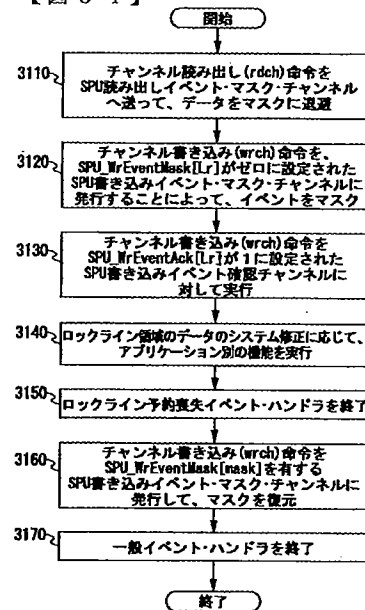
【図 62】



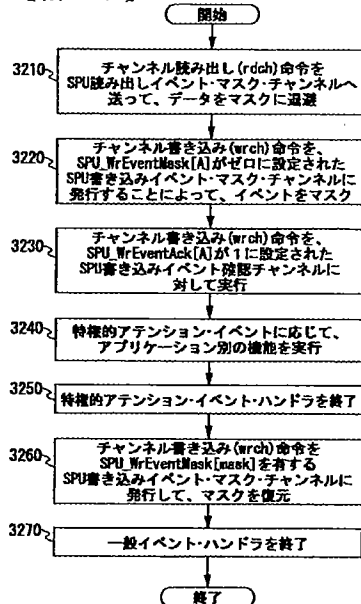
【図 6 3】



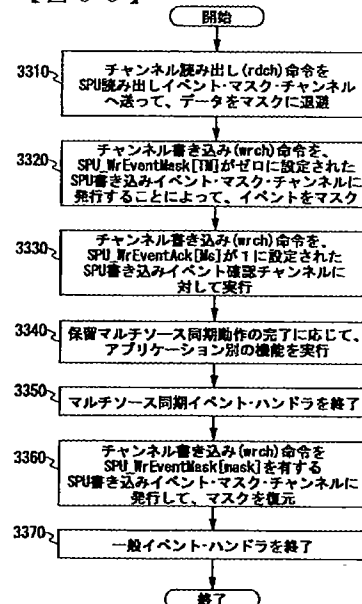
【図 6 4】



【図 6 5】



【図 6 6】



---

フロントページの続き

(74)代理人 100086243

弁理士 坂口 博

(72)発明者 チャールズ・レイ・ジョンズ

アメリカ合衆国 78759、テキサス州、オースティン、カッシア・ドライブ 10703

(72)発明者 トッド・スワンソン

アメリカ合衆国 78664、テキサス州、ラウンド・ロック、グヌー・ギャップ 1709

(72)発明者 ペイチュン・ピーター・リウ

アメリカ合衆国 78750、テキサス州、オースティン、リモンシージョ・ドライブ 9220

(72)発明者 ミシェル・ノーマン・デイ

アメリカ合衆国 78681、テキサス州、ラウンド・ロック、メイフィールド・ドライブ 2201

(72)発明者 トウオン・クアン・トロング

アメリカ合衆国 78727、テキサス州、オースティン、ピケット・ロープ・レーン 12612